

10-5-0 A
Please type a plus sign (+) inside this box →

PTO/SB/05 (4/98)
Approved for use through 09/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))

Attorney Docket No. AGLE0003

First Inventor or Application Identifier Calderone et al.

Title System and Method of a Multi-Dimensional Plex...

Express Mail Label No. EL540886848US

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. * Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)
2. Specification [Total Pages 115]
 - Descriptive title of the Invention
 - Cross References to Related Applications
 - Statement Regarding Fed sponsored R & D
 - Reference to Microfiche Appendix
 - Background of the Invention
 - Brief Summary of the Invention
 - Brief Description of the Drawings (if filed)
 - Detailed Description
 - Claim(s)
 - Abstract of the Disclosure
3. Drawing(s) (35 U.S.C. 113) [Total Sheets 47]
4. Oath or Declaration [Total Pages 2]
 - a. Newly executed (original or copy)
 - b. Copy from a prior application (37 C.F.R. § 1.63(d))
(for continuation/divisional with Box 16 completed)
 - i. DELETION OF INVENTOR(S)
Signed statement attached deleting
inventor(s) named in the prior application,
see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).

NOTE FOR ITEMS 1 & 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).

16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment

Continuation Divisional Continuation-in-part (CIP) of prior application No _____

Prior application information Examiner _____

Group / Art Unit _____

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. CORRESPONDENCE ADDRESS

<input checked="" type="checkbox"/> Customer Number or Bar Code Label	22862	or <input type="checkbox"/> Correspondence address below <i>(Insert Customer No. or Attach bar code label here)</i>
Name		
Address		
City	State	Zip Code
Country	Telephone	Fax

Name (Print/Type)	Christopher Peil	Registration No. (Attorney/Agent)	45,005
Signature	<i>Christopher Peil</i>	Date	10/4/2000

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL

for FY 1999

Patent fees are subject to annual revision.

Small Entity payments must be supported by a small entity statement, otherwise large entity fees must be paid. See Forms PTO/SB/09-12. See 37 C.F.R. §§ 1.27 and 1.28.

TOTAL AMOUNT OF PAYMENT (\$ 1,043.00)

Complete if Known

Application Number	Unassigned
Filing Date	Herewith
First Named Inventor	Calderone et al.
Examiner Name	Unassigned
Group / Art Unit	Unassigned
Attorney Docket No.	AGLE0003

METHOD OF PAYMENT (check one)

1. The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number 07-1445

Deposit Account Name Michael A. Glenn

 Charge Any Additional Fee Required
Under 37 CFR §§ 1.16 and 1.17

2. Payment Enclosed:

 Check Money Order Other

FEE CALCULATION (continued)

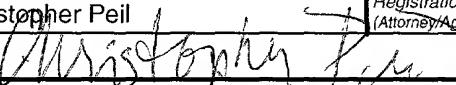
3. ADDITIONAL FEES

Large Entity Small Entity

Fee Code (\$)	Fee Code (\$)	Fee	Fee	Fee	Fee Description	Fee Paid
105	130	205	65		Surcharge - late filing fee or oath	
127	50	227	25		Surcharge - late provisional filing fee or cover sheet	
139	130	139	130		Non-English specification	
147	2,520	147	2,520		For filing a request for reexamination	
112	920*	112	920*		Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*		Requesting publication of SIR after Examiner action	
115	110	215	55		Extension for reply within first month	
116	380	216	190		Extension for reply within second month	
117	870	217	435		Extension for reply within third month	
118	1,360	218	680		Extension for reply within fourth month	
128	1,850	228	925		Extension for reply within fifth month	
119	300	219	150		Notice of Appeal	
120	300	220	150		Filing a brief in support of an appeal	
121	260	221	130		Request for oral hearing	
138	1,510	138	1,510		Petition to institute a public use proceeding	
140	110	240	55		Petition to revive - unavoidable	
141	1,210	241	605		Petition to revive - unintentional	
142	1,210	242	605		Utility issue fee (or reissue)	
143	430	243	215		Design issue fee	
144	580	244	290		Plant issue fee	
122	130	122	130		Petitions to the Commissioner	
123	50	123	50		Petitions related to provisional applications	
126	240	126	240		Submission of Information Disclosure Stmt	
581	40	581	40		Recording each patent assignment per property (times number of properties)	
146	760	246	380		Filing a submission after final rejection (37 CFR § 1.129(a))	
149	760	249	380		For each additional invention to be examined (37 CFR § 1.129(b))	
Other fee (specify) _____						
Other fee (specify) _____						
Reduced by Basic Filing Fee Paid						SUBTOTAL (3) (\$ 40.00)

SUBMITTED BY

Complete (if applicable)

Name (Print/Type)	Christopher Peil	Registration No. (Attorney/Agent)	45,005	Telephone	650-474-8400
Signature				Date	10/4/2000

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

**STATEMENT CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) & 1.27(c))--SMALL BUSINESS CONCERN**
Docket Number (Optional)
AGLE0003

Applicant, Patentee, or Identifier: Calderone et al.

Application or Patent No.: Unassigned

Filed or Issued: Herewith

Title: System and Method of a Multi-Dimensional Plex Communication Network

I hereby state that I am

the owner of the small business concern identified below:
 an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN AgileTV Corporation

ADDRESS OF SMALL BUSINESS CONCERN 333 Ravenswood Avenue, Bldg. 202
Menlo Park, CA 94025

I hereby state that the above identified small business concern qualifies as a small business concern as defined in 13 CFR Part 121 for purposes of paying reduced fees to the United States Patent and Trademark Office. Questions related to size standards for a small business concern may be directed to: Small Business Administration, Size Standards Staff 409 Third Street, SW, Washington, DC 20416.

I hereby state that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

the specification filed herewith with title as listed above.
 the application identified above.
 the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern, or organization having rights in the invention must file separate statements as to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern, or organization having any rights in the invention is listed below:
 no such person, concern, or organization exists.
 each such person, concern, or organization is listed below

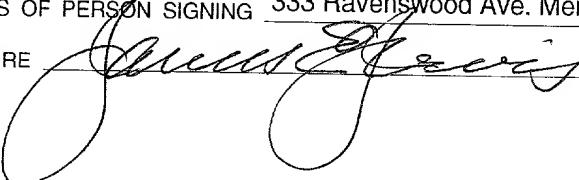
Separate statements are required from each named person/concern or organization having rights to the invention stating their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

NAME OF PERSON SIGNING James E. Jervis

TITLE OF PERSON IF OTHER THAN OWNER Vice President of Intellectual Property

ADDRESS OF PERSON SIGNING 333 Ravenswood Ave. Menlo Park, CA 94025

SIGNATURE 

DATE 10-3-00

SYSTEM, METHOD, AND NODE OF A MULTI-DIMENSIONAL PLEX COMMUNICATION NETWORK AND NODE THEREOF

5

Technical field

This invention relates to concurrent high-speed communication networks acting with multi-dimensional arrays of communications processors and nodes of such networks.

10

Background Art

Processors have long been coupled in various network configurations to enhance processing speed, processing power and processor intercommunication. Many such coupling arrangements sacrifice speed as the number of nodes of the processor network increases in size. Other arrangements couple all or nearly all nodes of the processor network to one another increasing speed at the expense of each node requiring substantial hardware and management expense. Still further prior art arrangements employ high speed switches interconnecting all network nodes to each other. The switches themselves become complex entities as the number of network nodes increase. Details of these prior art arrangements are discussed in the following section.

Figure 1A depicts two processors 100 and 110 coupled 106 to each other, each with accessibly coupled 102 and 112 memories 104 and 114, respectively, as found in the prior art.

Systems such as this can be found in many prior art settings. They are so common that the Windows NT™ operating system supports them. Many such operating systems allocate high-speed communication to one processor and use the other processor for more computationally intensive tasks, such as image processing algorithms. Operating systems often provide real-time event-driven software tools to aid in the control of such computing systems. Operating systems often further provide real-time event-driven software tools supporting message passing to organize communication between concurrent tasks or objects, which may reside in different processors.

Processor 100 is accessibly coupled 102 to memory 104. Processor 110 is accessibly coupled 112 to memory 114. Processor 100 is communicatively coupled 106 to processor 110. Processor 110 is further coupled 116 with an external communications network.

Typically, processor 110 takes care of the large IO overhead tasks, such as communicating 116 with a high-speed network. Processor 100 is often allocated to perform one or more large local computational tasks, as often found in graphics development, signal processing, and image processing applications.

Note that in many situations, each processor would be seen as a node in the communications scheme. In many other situations, the combination of the two processors would be seen as a node in a larger computing network.

Figure 1B depicts a communication scheme through a square array of 16 nodes,

5 as found in the prior art.

Consider the transfer of a message from one node to another node. The communication performance is often measured in terms of hops. A hop can be seen as the number of arrows encountered when going from the first node to the second node. Such communication schemes have the problem of requiring a

10 very large number of hops to communicate between certain nodes. In this scheme, the number of hops can be half the total number of nodes in the communication scheme. In Figure 1B, assume N=4 nodes in each of the two orthogonal directions (up-down and left-right). The maximum number of hops is half the total number of nodes, or $(N^2)/2$ hops or 8 hops.

15 Figure 1C depicts a two-dimensional communications mesh architecture coupled with the two-dimensional node array of 4 rows and 4 columns of Figure 1B, as found in the prior art.

Note that as used herein, N will refer to the number of nodes in each orthogonal direction, or side of a multi-dimensional array of nodes. The dimensionality of the

20 array will be referred to as M. When a node array is not uniform in each orthogonal direction, it will be specifically noted.

Figure **1C** depicts a nearest neighbor communications scheme based around a two dimensional rectangular grid arrangement of nodes **200**. This has been extensively researched and applied in a variety of large scale applications since at least the 1980's.

5 The great strength of this scheme is that it provides an interconnect mechanism which is a big improvement over connecting the same number of processors in a the communication of Figure **1B**, while requiring only a relatively small number of interconnects.

The maximum number of hops necessary to communicate between two

10 arbitrarily selected nodes is the sum of the arrows on the sides of the array, or $2*(N-1)$. The scheme requires more hops as either the number of rows or columns increase, albeit at a better rate than the communication scheme of Figure **1B**.

Figure **2A** depicts a two-dimensional, 2-D, toroidal mesh communications
15 architecture of three rows and three columns, as found in the prior art.

A 2-D mesh as shown in Figure **1C** can be viewed as residing on a 2-D surface of a piece of paper.

Figure **2B** depicts the prior art two-dimensional, 2-D, toroidal mesh communications architecture of three rows and three columns shown in Figure
20 **2A** but drawn on a torus.

A 2-D toroidal mesh is viewed as residing on the surface of a torus, formed by connecting the left side and right side and then connecting the top side and bottom side, of a 2-D mesh as shown in Figure 2B. The advantage of the toroidal architecture is that it reduces the longest travel time through the 5 communication network to about half that of the 2-D mesh travel time in terms of hops.

3-D toroidal mesh architectures have the same basic properties, although they are much more difficult to visualize. They assume a 3-D locally rectangular lattice of neighboring nodes, but are much more difficult to visualize.

10 In a 2-D toroidal scheme, the communications paths are inherently planar, and in cases above three nodes to a row or column, do not provide complete interconnect between the nodes of a row or column. In point of fact, they teach a locally planar connection grid which, above three nodes in each orthogonal direction, do not form a complete interconnect between the nodes of any 15 orthogonal direction. What is desired is a communications scheme overcoming the limitations of the existing nearest neighbor multi-dimensional communications schemes when N is larger than three. Several schemes have evolved to this end.

20 Figure 2C depicts eight nodes with a total interconnect communications grid, with every node directly coupled to each of the other nodes, as found in the prior art. Note that each of these nodes must have seven ports to support the communications scheme. So for N^2 nodes in a square configuration, where N is 2 or larger, each node would need N^2-1 ports. This is a serious complexity

burden not only on the port hardware at the node, but also in terms of communications management needed at each node to control port communications. This can be a major problem. What is further desired is a communications scheme, which does not place such severe requirements on 5 each node, but supports nearly the same communication performance.

Figure 3A depicts eight nodes communicating directly with a switch 290 and through the switch, each node is directly coupled to each of the other nodes by means of two hops, as found in the prior art.

These systems provide a communication path from each node to any other node

10 in just two hops, one to the switch from the source node and a second from the switch to the destination node. Such interconnect schemes are often chosen today. They provide fast interconnect, but the complexity of the communication switching rapidly dwarfs the complexity of the rest of the system as the number of nodes increase.

15 Communication protocol switches today tend to be either Ethernet to Ethernet switch architectures or Ethernet to ATM back to Ethernet switch architectures.

The first is often a circuit switched approach, sometimes involving cross bar switches. The second is a packet switch approach, using asynchronous traversal of the ATM network to wormhole packets from their source to their 20 destinations. These switches are inherently complex. This complexity has negative impact on the initial cost, maintenance, and reliability of such switches and their systems.

Consider the common communications scheme of a 64 port Ethernet switch interconnecting 64 nodes. Such switches are extremely complex. The switch complexity is far greater than the rest of the system taken as a whole. The switch complexity dominates the cost, maintenance and reliability of everything else in such systems. What is further desired is a fast, minimal overhead, lower cost, communications scheme which provides nearly the same performance in terms of hops, but at a fraction of the complexity of the system as a whole.

Figure 3B depicts a four dimensional hypercube of 2^4 nodes as found in the prior art.

10 In this example, M, the dimension of the array, is four and N, the number of nodes in an orthogonal direction, is two. Each of the nodes has M=4 ports. A communication between nodes can take up to M hops. The total interconnects of the system is $(M^*N^M)/2$, which rapidly exceeds the number of processors. This interconnect scheme is a nearest orthogonal neighbor scheme, which when N is
15 larger than two, shares the same problems as the mesh architectures.

The hypercube interconnect architecture points to a basic trend in the parallel processor community, the necessity of rewriting major software programs into specialized parallel processor programs. These rewritten parallel processor programs distribute the data processing to be performed over many processing
20 units, which then must communicate their computational results or decisions to at least some other of the processors.

There are numerous schemes for controlling these activities within a node, Single Instruction Multiple Datapath (SIMD), and Multiple Instruction Multiple Datapath (MIMD) are just two approaches. SIMD architectures concurrently act on a single instruction across multiple datapaths. MIMD architectures concurrently act on multiple instructions across multiple datapaths. Note that Single Instruction Single Datapath (SISD) architectures act on a single instruction across one datapath and include microprocessors. Multiple Instruction Single Datapath (MISD) architectures concurrently act on multiple instructions across a single datapath and include MPEG stream decoders.

- 5 Almost all programs are initially written for SISD architectures. Rewriting major programs for data processing on these various alternative parallel processor architectures often requires reinventing the underlying algorithms of those programs in a concurrent form for the various control and communications schemes.
- 10 By way of example, people providing single processor weather prediction or air-frame simulations have built computational tools often at the conceptual limits of comprehension and verification. Such tools do not translate readily into these alternative computer architectures and usually require great effort just to get the algorithms to run in these new environments, much less improve their results.
- 15 What is further desired is a communication scheme to enable many processors to support major computational problems with existing software, while requiring only minor software conversion.
- 20

Figure 3C depicts the use of two optical fibers to create a bidirectional communications physical transport directly connecting through taps to each of the nodes through a control point as taught by the prior art multiplexing multiple signals into a single physical transport.

5 This use of fiber optics can be found in Figure 8 and its discussion in U.S. Patents 5,029,962 and 5,037,170. This use of fiber optics can be found in Figure 4 and its discussion in U.S. Patent 4,938,008. This use of fiber optics can be found in Figure 3 and its discussion in U.S. Patent 4,889,403. This use of fiber optics can be found in Figure 1 and its discussion in U.S. Patent 4,815,805. This
10 use of fiber optics can be found in Figures 4, 76-78 and its discussion in U.S. Patent 4,768,854. Note that this patent discusses use of a third optical fiber. This use of fiber optics can be found in Figure 8 and its discussion in U.S. Patents 4,741,585 and 4,824,199.

It is common in the above-cited prior art to discuss the control point as a
15 “headend” which distinguishes the signals received from the nodes using a photocell generating an electrical signal. The electrical signal is tested for a “1” or a “0” condition.

Figure 3D depicts the use of one optical fiber to create a bidirectional communications physical transport directly connecting through taps to each of
20 the nodes through a control point as taught by the prior art multiplexing multiple signals into a single physical transport.

This use of fiber optics can be found in Figure 3 and its discussion in U.S. Patents 4,822,125 and 4,557,550.

Figure 3E depicts the use of one optical fiber to create a uni-directional communications physical transport directly connecting through taps to each of 5 the nodes through a control point as taught by the prior art distributing a collection of multiple signals into a single physical transport for multiple destinations.

This use of fiber optics can be found in Figure 5 and its discussion in U.S. Patents 4,747,652 and 4,834,482.

10 Figures 3C, 3D and 3E have been shown to present various prior art methods employing the physical transport of one or more optical fibers to multiplex and/or distribute multiple communications between multiple points.

To summarize the shortcomings of the prior art, what is needed is a 15 communications scheme supporting nearly the same communication performance while placing much less severe requirements on each node than the complete direct interconnect approach. What is also needed is a fast, minimal efficient communications scheme providing performance similar to the ideal switch interconnect approach, but at a fraction of the complexity of the system as a whole.

Summary of the Invention

Certain embodiments of the invention advantageously provide communications performance similar to the switch interconnect approach, and nearly the same communications performance as the complete direct interconnect approach but at dramatically reduced complexity compared to other known approaches having similar communications performance.

The communications network has M orthogonal directions that support communications between an M dimensional lattice of up to N^M nodes, where M is at least two and N is at least four. Each node pencil in a first orthogonal

10 direction contains at least four nodes and each node pencil in a second orthogonal direction contains at least two nodes. Each of the nodes contains a multiplicity of ports.

As used herein, a node pencil refers to a 1-dimensional collection of nodes differing from each other in only one dimensional component, i.e. the orthogonal

15 direction of the pencil. By way of example, a nodal pencil in the first orthogonal direction of a two-dimensional array contains the nodes differing in only the first dimensional component. A nodal pencil in the second orthogonal direction of a two-dimensional array contains the nodes differing in only the second dimensional component. Node(a,b) will refer to a node in the a location in the first 20 orthogonal direction and b location in the second orthogonal direction.

The communications network is comprised of a communication grid interconnecting the nodes. The communications grid includes up to $N^{(M-1)}$

communication pencils, for each of the M directions. Each of the communication pencils in each orthogonal direction corresponds to a node pencil containing a multiplicity of nodes, and couples every pairing of nodes of the node pencil directly.

5 As used herein, communication between two nodes of a nodal pencil coupled with the corresponding communication pencil comprises traversal of the physical transport layer(s) of the communication pencil.

Such embodiments of the invention advantageously support direct communication between any two nodes belonging to the same communication

10 pencil, supporting communication between any two nodes in an M dimensional array in at most M hops.

Comparing the invention to existing, prior art direct connection of all nodes finds the following. Direct connection of all nodes provides communication between

nodes in one hop, but requires each node have almost as many ports as there

15 are nodes in the array. Each of these ports adds complexity not only to node hardware, but also to the node-resident management of the node's port communication.

Each node pencil in the second orthogonal direction may contain at least three nodes. Each node pencil in the second orthogonal direction may contain at least

20 four nodes.

Communication between nodes utilizing a communications switch, provides communication between nodes in two hops, but finds the communication switch

complexity rapidly dominating the entire system, adversely affecting initial capital expenditure and maintenance of the system. The reliability of such systems is jeopardized, in that if the switch fails, the system fails.

Each of the communication pencils may be comprised of the number of

5 communications paths required to interconnect each node of the corresponding

node pencil directly to the other nodes of the corresponding node pencil. Such

embodiments of the invention advantageously support communications paths

along each communication pencil based upon point-to-point physical transport

layers of various wireline structures, such as wire, fiber optics, twisted pair,

10 coaxial cable, wave-guides such as micro-channel and free space lasers. Free

space lasers essentially operate without a physical wireline, but are

fundamentally directed through free space in a fashion closer to wireline than

wireless physical transports. They will for the sake of clarity of discourse be

considered a wireline physical transport herein. Communication path support

15 may further include, but is not limited to, Wavelength Division Multiplexing

(WDM). Communication pencils may advantageously support Ethernet, ArcNet,

Token Ring, FDDI and ATM as link layers. The communication grid may

advantageously support network layers including, but not limited to, TCP/IP,

Netware IPX, SMB and DecNet.

20 Note that, if a node fails in the communication grid, communication between any

pair nodes not including the failed node occurs at almost the same efficiency.

If a coupling of a node to a communication pencil fails, or if a communication

path between two nodes fails in one communication pencil, the system can route

communication between any two nodes through different pencils such that communications performance is not lost. In the same way, if one node fails, the communication between two functioning nodes is at most M hops by rerouting communication through functioning nodes.

- 5 Certain embodiments of the invention include communicating between a first node and a second node. The first node is coupled to a first communication pencil coupled to a third node. The third node is coupled a second communication pencil coupled to the second node. Each of the communication pencils includes at least one physical transport layer.
- 10 Communicating between the first node and second node includes the following. The first node communicates with the third node via the first communication pencil by traversing all of the physical transport layers included in the first communication pencil. The third node communicates with the second node via the second communication pencil by traversing all of the physical transport layers
- 15 included in the second communication pencil.

Such embodiments advantageously support extremely high speed communication through the communication pencils, traversing the physical transport layers and using intermediate third nodes, when direct communication between the first node and second node is either excessively costly or infeasible

- 20 for other reasons.

Certain embodiments of the invention include a node coupling to M communication pencils, where M is at least two. The node includes M

communication interfaces, each of the communication interfaces coupling to a corresponding communication pencil. The node support a communication process performed within the node controlling all of the communications interfaces. The communication process includes interacting within the node with

5 the communication interface, for each of the communication interfaces.

Each of these interactions within the node further include the following. Receiving a first communication from the communication interface to create a received communication from the communication interface. Processing the received communication from the communication interface. And sending a local

10 communication to the communication interface to create a second communication to the communication interface.

Such embodiments advantageously support interactive communication control within the node controlling the communication interfaces to each communication pencil coupled to the node.

15 Note that an appendix listing a C programming language model of an embodiment of the invention is attached to this document. It will be apparent to one of skill in the art that this model shows decisions being weighed regarding communicating within a node, communicating external to a node, communicating through the node to other communication interfaces, communicating through the

20 communication interfaces to communicate to elements within the node, communicating through tunnel interfaces, communicating through one or more communication processor coupling mechanism, communicating based upon avoidance of obstructions and communicating based upon various cost factors.

Obstructions may include, but are not limited to, various systems failures, omitted elements, network management allocations including but not limited to network partitioning, database access privileges, and network access privileges. Cost factors may include but are not limited to overall communication delay, speed, 5 bandwidth utilization and node resource utilization. Note that obstructions may be expressed as cost factor with exorbitant costs.

Note that the appendix is a working simulation of an embodiment of the invention, and as such, represents an actual reduction to practice of the invention. The model presented in the appendix is just one of many 10 embodiments and has been included to meet in part the duty of candor and demonstrate the details of that implementation. This appendix and the embodiment it models are in no way meant to limit the scope of the claims.

These and other advantages of the present invention will become apparent upon reading the following detailed descriptions and studying the various figures of the 15 drawings.

Brief Description of the Drawings

Figure 1A depicts two processors coupled to each other, each with accessibly coupled memories as found in the prior art;

20 Figure 1B depicts a communication scheme through a square array of 16 nodes, as found in the prior art;

Figure **1C** depicts a two-dimensional communications mesh architecture coupled with the two-dimensional node array of 4 rows and 4 columns of Figure **1B**, as found in the prior art;

Figure **2A** depicts a two-dimensional, 2-D, toroidal mesh communications architecture of three rows and three columns, as found in the prior art;

Figure **2B** depicts the prior art two-dimensional, 2-D, toroidal mesh communications architecture of three rows and three columns shown in Figure **2A** but drawn on a torus;

Figure **2C** depicts eight nodes with a total interconnect communications grid, with every node directly coupled to each of the other nodes, as found in the prior art;

Figure **3A** depicts eight nodes communicating directly with a switch **290** and through the switch, each node is directly coupled to each of the other nodes by means of two hops, as found in the prior art;

Figure **3B** depicts a four dimensional hypercube of 2^4 nodes as found in the prior art;

Figure **3C** depicts the use of two optical fibers to create a bidirectional communications physical transport directly connecting through taps to each of the nodes through a control point as taught by the prior art multiplexing multiple signals into a single physical transport;

Figure **3D** depicts the use of one optical fiber to create a bidirectional communications physical transport directly connecting through taps to each of

the nodes through a control point as taught by the prior art multiplexing multiple signals into a single physical transport;

Figure 3E depicts the use of one optical fiber to create a uni-directional communications physical transport directly connecting through taps to each of the nodes through a control point as taught by the prior art distributing a collection of multiple signals into a single physical transport for multiple destinations;

Figure 4A depicts a system 600 including a two-dimensional plex communication grid comprised of communication pencils 400, 410, 420 and 430 in the first orthogonal direction and communication pencils 300, 310, 320 and 330 in the second orthogonal direction, each with N=4 nodes 500, in accordance with certain embodiments;

Figure 4B depicts the system 600 including the two-dimensional plex communication grid of Figure 4A with highlighted communication pencils 400, 410, 420 and 430 in the first orthogonal direction;

Figure 4C depicts the system 600 including the two-dimensional plex communication grid of Figure 4A with highlighted communication pencils 300, 310, 320 and 330 in the second orthogonal direction;

Figure 4D depicts a system 600 including a two-dimensional plex communication grid with N=4 nodes 500, each node 500 containing six ports in accordance with certain embodiments of the invention;

Figure 5 depicts a system 600 including a two-dimensional plex communication grid with N=4 nodes 500, each node containing six ports, two communications processors, each coupled to three ports in accordance with certain embodiments of the invention;

5 Figure 6 depicts the two communications pencils 410 and 320 coupled to Node 2,1 (500) and the two node pencils of Node 2,1 of Figure 4D in accordance with certain embodiments of the invention;

Figure 7 depicts the communications pencils and their coupling to the node pencils in the first orthogonal direction of the two-dimensional plex communications grid of Figure 4D in accordance with certain embodiments of the invention;

10 Figure 8 depicts the communications pencils and their coupling to the node pencils in the second orthogonal direction of the two-dimensional plex communications grid of Figure 4D in accordance with certain embodiments of the invention;

15 Figure 9 depicts the communications pencils 410 and 320 coupled to Node 2,1 (500) highlighting communication pencil 410 in the first orthogonal direction of Figures 4D and 6;

Figure 10 depicts the communications pencils 410 and 320 coupled to Node 2,1 (500) highlighting communication pencil 320 in the second orthogonal direction of Figures 4D and 6;

Figure 11 depicts the communications pencils **410** and **320** coupled to Node 2,1 (500) highlighting the node pencil in the first orthogonal direction of Figures **4D** and **6**;

Figure 12 depicts the communications pencils **410** and **320** coupled to Node 2,1

5 highlighting the node pencil in the second orthogonal direction of Figures **4D** and **6**;

Figure 13A depicts the difference between the embodiments of the invention depicted in Figure **4D** of a 2-D, $N=4$ plex communications grid and the communications grid of a 2-D toroidal mesh interconnecting an N by N grid of nodes as shown in Figure **1C**;

10 Figure 13B depicts a 2-D, $N=4$ toroidal grid of the prior art mapped onto a torus, in a fashion similar to Figure **2B**;

Figure 13C depicts the difference between the embodiments of the invention depicted in Figure **4D** of a 2-D, $N=4$ plex communications grid and the communications grid of a 2-D toroidal mesh interconnecting an N by N grid of nodes as shown in Figure **13B**;

15 Figure 14 depicts a system **700** including a three-dimensional array of $N=5$ nodes **500** with orthogonal directions **490**, **492** and **494** in accordance with the invention;

Figure 15 depicts some of the node pencils and corresponding communications pencils in the first orthogonal direction of a three-dimensional array of 4*5*5 nodes **500** in accordance with certain embodiments of the invention;

Figure 16 depicts some of the node pencils and corresponding communications

5 pencils in the second orthogonal direction of a three-dimensional array of 4*5*5 nodes **500** in accordance with certain embodiments of the invention;

Figure 17 depicts some of the node pencils and corresponding communications

3 pencils in the third orthogonal direction of a three-dimensional array of 4*5*5

nodes **500** in accordance with certain embodiments of the invention;

10 Figure 18A depicts a node **500** with up to $M*(N-1)$ ports which couple to the communication pencils of the node which intersect at the node in an M -dimensional array in accordance with certain embodiments of the invention;

Figure 18B depicts a node **500** with up to P CPU's **510, 520, 550** and **560**

coupled by **530-536** to the communication pencils of node **500** which intersect at

15 node **500** in an M -dimensional array in accordance with certain embodiments of the invention; and

Figure 19 depicts a node **500** with up to P CPU's **510, 520, 550** and **560** directly

coupled by **538-548**, with ports **516, 526, 556** and **566** which couple to the

communication pencils of the node pencils intersecting at the node **500** in an M -

20 dimensional array in accordance with certain embodiments of the invention;

Figure 20A depicts a system 600 containing an M=2, N=4 plex communication grid with $16=N^M$ nodes 500, with one node 500 including four communication processing units (CPUs), three nodes 500 including three CPUs, and 12 nodes 500 including two CPUs, where the 3 CPU nodes each are further coupled to external communications networks and each possess a tunneling interface, in accordance with certain embodiments of the invention;

5 Figure 20B alternatively depicts a system 600 containing an M=2, N=4 plex communication grid with $16=N^M$ nodes 500, with one node 500 including four communication processing units (CPUs), three nodes 500 including three CPUs, and 12 nodes 500 including two CPUs, where the 3 CPU nodes each are further coupled to external communications networks and each possess a tunneling interface, in accordance with certain embodiments of the invention;

10 Figure 21 depicts a system 800 including two instances of system 600 as depicted in Figures 20A and 20B, referred to as 600-1 and 600-2, in accordance with certain embodiments of the invention;

15 Figure 22 partially depicts a toroidal three-dimensional mesh communication grid for an N=3, three dimensional (M=3) array of nodes 500, each comprising P=2 Communication Processor Units (CPU) which each comprise M=3 ports for the corresponding communication pencils;

20 Figure 23A depicts a flowchart of a method communicating between a first node 500 and a second node 500 when the first node 500 and second node 500 are

both coupled to communication pencils coupling to a third node **500**, in accordance with certain embodiments of the invention;

Figure **23B** depicts a detail flowchart of operation **1004** of Figure **23A** further performing the first node communicating with the third node via the first communication pencil;

Figure **23C** depicts a detail flowchart of operation **1008** of Figure **23A** further performing the third node communicating with the second node via the second communication pencil;

Figure **24A** depicts a detail flowchart of operation **1032** of Figure **23B** further performing traversing the physical transport layers when the first communication pencil includes a first physical transport layer and second physical transport layer;

Figure **24B** depicts a detail flowchart of operation **1052** of Figure **24C** further performing traversing the physical transport layers when the communication pencil includes a first physical transport layer and second physical transport layer;

Figure **25** depicts a flowchart of a communication process performed within the node **500** controlling all of the M communications interfaces, where M is between 2 and 5, in accordance with certain embodiments of the invention;

Figure 26 depicts a detail flowchart of operation 1304 of Figure 25 further performing interacting within the node with the first communication interface in accordance with certain embodiments of the invention;

Figure 27A depicts a detail flowchart of operation 1372 of Figure 26 further

5 performing for each of the communication interfaces coupled to the communication processor, receiving the first communication;

Figure 27B depicts a detail flowchart of operation 1382 of Figure 26 further

performing for each of the communication interfaces coupled to the communication processor, processing the received communication;

10 Figure 27C depicts a detail flowchart of operation 1392 of Figure 26 further performing for each of the communication interfaces coupled to the communication processor, sending the local communication;

Figure 28A depicts a detail flowchart of operation 1432 of Figure 27B further

15 performing processing the received communication in accordance with certain embodiments of the invention;

Figure 28B depicts an alternative detail flowchart of operation 1432 of Figure

27B further performing processing the received communication in accordance with certain embodiments of the invention;

Figure 29A depicts a detail flowchart of operation 1512 of Figure 28A further

20 performing determining the received communication destination;

Figure 29B depicts an alternative detail flowchart of operation 1512 of Figure 28A further performing determining the received communication destination in accordance with certain embodiments of the invention;

Figure 30A depicts a detail flowchart of operation 1300 of Figure 25 further 5 performing the communication process within the node;

Figure 30B depicts a detail flowchart of operation 1582 of Figure 29A further performing evaluating the destination component;

Figure 31A depicts a detail flowchart of operation 1522 of Figure 29A further 10 performing routing the received communication for a communication processor coupled to the communication processor coupling mechanism;

Figure 31B depicts a detail flowchart of operation 1522 of Figure 29A further performing for each of the communication processors, routing the received communication;

Figure 31C depicts a detail flowchart of operation 1532 of Figure 28A further 15 performing for each of the communication processors, delivering the received communication;

Figure 32A depicts a detail flowchart of operation 1532 of Figure 28A further performing for each of the communication processors, the step of delivering the received communication;

Figure 32B depicts a detail flowchart of operation 1432 of Figure 27B further performing processing the received communication from the communication interface coupled to the communication processor;

Figure 33A depicts a detail flowchart of operation 1712 of Figure 32B further

5 performing determining based upon the received communication from the communication interface;

Figure 33B depicts a detail flowchart of operation 1612 of Figure 30A further

performing maintaining the routing table;

Figure 34A depicts a detail flowchart of operation 1782 of Figure 33B further

10 performing distributing the new routing table;

Figure 34B depicts a detail flowchart of operation 1792 of Figure 34A further

performing communicating the new routing table;

Figure 35A depicts a detail flowchart of operation 1802 of Figure 34A further

performing replacing the routing table;

15 Figure 35B depicts a detail flowchart of operation 1772 of Figure 33B further

performing generating the new routing table;

Figure 36A depicts a detail flowchart of operation 1532 of Figure 28A further

performing delivering the received communication;

Figure 36B depicts a detail flowchart of operation 1532 of Figure 28A further

20 performing delivering the received communication;

Figure 36C depicts a detail flowchart of operation 1382 of Figure 26 further performing processing the received communication;

Figure 37A depicts a detail flowchart of operation 1932 of Figure 36C further performing assessing the received communication;

5 Figure 37B depicts an alternative detail flowchart of operation 1932 of Figure 36C further performing assessing the received communication; and

Figure 38 depicts a communication interface 900 including P1=4 input ports and P2=4 output ports coupled to a communication pencil including optical fibers 902 and 904, each optical fiber handling one way traffic with optical fiber 902 coupled 10 940 through optronic amplifier 942 coupling 944 to optical fiber 904, in accordance with certain embodiments of the invention.

Detailed Description of the Invention

Figure 4A depicts a system 600 including a two-dimensional plex communication 15 grid comprised of communication pencils 400, 410, 420 and 430 in the first orthogonal direction and communication pencils 300, 310, 320 and 330 in the second orthogonal direction, each with N=4 nodes 500, in accordance with certain embodiments of the invention.

Note that M is two, the dimension of the array of nodes. N is four and each row 20 and column, or node pencil in either of the two orthogonal directions, has 4 nodes.

The communications network is comprised of a communication grid interconnecting the nodes. The communications grid includes up to $N^{(M-1)}$, or 4 communication pencils, for each of the $M=2$ orthogonal directions. Each of the communication pencils in each orthogonal direction is coupled with a corresponding node pencil containing a multiplicity of nodes coupling every pairing of nodes of the corresponding node pencil directly.

Communication between two nodes **500** of a nodal pencil coupled with the corresponding communication pencil includes traversal of the physical transport layer(s) of the communication pencil. Such embodiments of the invention advantageously support communications paths along each communication pencil based upon point-to-point physical transport layers, such as fiber optics, microwave wave guides such as micro-channel, and free space lasers. Further embodiments of the invention support Wavelength Division Multiplex (WDM) through the physical transport of the communication paths of the communication pencils.

These communications pencils advantageously support communications paths based upon point-to-point physical transport layers of various wireline structures, such as wire, fiber optics, twisted pair, coaxial cable, wave-guides such as micro-channel and free space lasers. Communication path support may include, but is not limited to, Wavelength Division Multiplexing (WDM). Communication pencils may advantageously support Ethernet, ArcNet, Token Ring, FDDI and ATM as link layers. The communication grid may advantageously support network layers including, but not limited to, TCP/IP, Netware IPX, SMB and DecNet.

Figure 4B depicts the two-dimensional plex communication grid of Figure 4A with highlighted communication pencils 400, 410, 420 and 430 in the first orthogonal direction. Recall that a nodal pencil in the first orthogonal direction of a two-dimensional array contains the nodes 500 differing in only the first dimensional component.

Consider the node pencil in the first orthogonal direction containing Node 0,0, Node 1,0, Node 2,0 and Node 3,0. The communication pencil 400 in the first orthogonal direction couples to the nodes of this node pencil. Node 0,0 is coupled 402 to communication pencil 400. Node 1,0 is coupled 404 to communication pencil 400. Node 2,0 is coupled 406 to communication pencil 400. Node 3,0 is coupled 408 to communication pencil 400.

Consider the node pencil in the first orthogonal direction containing Node 0,1, Node 1,1, Node 2,1 and Node 3,1. The communication pencil 410 in the first orthogonal direction couples to the nodes of this node pencil. Node 0,1 is coupled 412 to communication pencil 410. Node 1,1 is coupled 414 to communication pencil 410. Node 2,1 is coupled 416 to communication pencil 410. Node 3,1 is coupled 418 to communication pencil 410.

Consider the node pencil in the first orthogonal direction containing Node 0,2, Node 1,2, Node 2,2 and Node 3,2. The communication pencil 420 in the first orthogonal direction couples to the nodes of this node pencil. Node 0,2 is coupled 422 to communication pencil 420. Node 1,2 is coupled 424 to communication pencil 420. Node 2,2 is coupled 426 to communication pencil 420. Node 3,2 is coupled 428 to communication pencil 420.

Consider the node pencil in the first orthogonal direction containing Node 0,3, Node 1,3, Node 2,3 and Node 3,3. The communication pencil **430** in the first orthogonal direction couples to the nodes of this node pencil. Node 0,3 is coupled **432** to communication pencil **430**. Node 1,3 is coupled **434** to communication pencil **430**. Node 2,3 is coupled **436** to communication pencil **430**. Node 3,3 is coupled **438** to communication pencil **430**.

Figure **4C** depicts the two-dimensional plex communication grid of Figure **4A** with highlighted communication pencils **300**, **310**, **320** and **330** in the second orthogonal direction. Recall that a nodal pencil in the second orthogonal direction of a two-dimensional array contains the nodes **500** differing in only the second dimensional component.

Consider the node pencil in the second orthogonal direction containing Node 0,0, Node 0,1, Node 0,2 and Node 0,3. The communication pencil **300** in the second orthogonal direction couples to the nodes of this node pencil. Node 0,0 is coupled **302** to communication pencil **300**. Node 0,1 is coupled **304** to communication pencil **300**. Node 0,2 is coupled **306** to communication pencil **300**. Node 0,3 is coupled **308** to communication pencil **300**.

Consider the node pencil in the second orthogonal direction containing Node 1,1, Node 1,1, Node 1,2 and Node 1,3. The communication pencil **310** in the second orthogonal direction couples to the nodes of this node pencil. Node 1,0 is coupled **312** to communication pencil **310**. Node 1,1 is coupled **314** to communication pencil **310**. Node 1,2 is coupled **316** to communication pencil **310**. Node 1,3 is coupled **318** to communication pencil **310**.

Consider the node pencil in the second orthogonal direction containing Node 2,2, Node 2,1, Node 2,2 and Node 2,3. The communication pencil **320** in the second orthogonal direction couples to the nodes of this node pencil. Node 2,0 is coupled **322** to communication pencil **320**. Node 2,1 is coupled **324** to communication pencil **320**. Node 2,2 is coupled **326** to communication pencil **320**. Node 2,3 is coupled **328** to communication pencil **320**.

Consider the node pencil in the second orthogonal direction containing Node 3,3, Node 3,1, Node 3,2 and Node 3,3. The communication pencil **330** in the second orthogonal direction couples to the nodes of this node pencil. Node 3,0 is coupled **332** to communication pencil **330**. Node 3,1 is coupled **334** to communication pencil **330**. Node 3,2 is coupled **336** to communication pencil **330**. Node 3,3 is coupled **338** to communication pencil **330**.

Each of the communication pencils may be comprised of the number of communications paths required to interconnect each node of the corresponding node pencil directly to the other nodes of the corresponding node pencil. Such embodiments of the invention advantageously support communications paths along each communication pencil based upon point-to-point physical transport layers, such as fiber optics, and microwave wave guides such as micro-channel.

These communications paths along each communication pencil advantageously support point-to-point physical transport layers of various wireline structures, such as wire, fiber optics, twisted pair, coaxial cable, wave-guides such as micro-channel and free space lasers. Communication path support may include, but is not limited to, Wavelength Division Multiplexing (WDM). Communication pencils

may advantageously support Ethernet, ArcNet, Token Ring, FDDI and ATM as link layers. The communication grid may advantageously support network layers including, but not limited to, TCP/IP, Netware IPX, SMB and DecNet.

Figure 4D depicts a two-dimensional plex communication grid with N=4 nodes

5 **500**, each node **500** containing six ports in accordance with certain embodiments of the invention. Note that M=2 and N=4. Each of the nodes **500** has $M^*(N-1)$, or six ports. Three of these ports on each node **500** are devoted to providing a direct interconnect to at least the other nodes **500** of its column through a collection of communication paths forming the communication pencil in the first 10 orthogonal direction. The nodes **500** belonging to the same column are the nodes **500** of the node pencil in the first orthogonal direction. The nodes **500** belonging to the same row are the nodes **500** of the node pencil in the second orthogonal direction.

15 Three of these ports on each node **500** are devoted to providing a direct interconnect to the other nodes **500** of its row through a collection of communication paths forming the communication pencil in the second orthogonal direction. Those nodes **500** belonging to the same row are the nodes **500** of the node pencil in the second orthogonal direction.

20 In further embodiments of the invention, at least one node **500** has at least one additional port. At least one of the additional ports may be connected to an external network. Further, at least one of the additional ports may be connected to an external mass storage system. In other embodiments of the invention, at

least one of the additional ports may be connected to an external database system.

A node **500** may contain at least one instruction processor. As used herein, an instruction processor includes but is not limited to instruction set processors, 5 inference engines and analog processors. An instruction set processor refers to instruction processors changing state directly based upon an instruction, and which change an internal state by executing the instruction. Note that the instruction may include, but is not limited to, direct or native instructions and interpreted instructions. An inference engine changes state when presented an 10 instruction, which may include a assertion, an assumption, or an inference rule. Inference engines include, but are not limited to, Horn clause engines such as Prolog requires, constraint based systems and neural network engines. As referred to herein, analog processors include, but are not limited to, optical signal 15 processors, CCD's, and resonant cavity devices responding to data and/or controls asserted in the analog domain.

Communication includes, but is not limited to, communication using a digital communications protocol. Communication also includes a messaging protocol using the digital communications protocol. Communications also includes a messaging protocol compatibly supporting TCP-IP, supporting the Internet, 20 and/or supporting the World Wide Web. Communications also includes link layers including but not limited to Ethernet, ArcNet, Token Ring, FDDI and ATM. Communication also includes network layers including, but not limited to, TCP/IP, Netware IPX, SMB and DecNet.

Communications may also include at least one video stream protocol using a digital communications protocol. In further embodiments of the invention, communications includes at least one multi-media stream protocol using the video stream protocols which may include motion JPEG and may also include at 5 least one form of MPEG.

Further embodiments of the invention support Wavelength Division Multiplex (WDM) through the physical transport of the communication paths of the communication pencils.

Each node may include a communication processor. Each node may further 10 include P communications processors. P may be a factor of number of communications ports required by the communications grid to couple with the node. The number of required ports may be $M^*(N-1)$, so that P becomes a factor of $M^*(N-1)$. Such embodiments of the invention advantageously support communications processing at the node partitioned across the P communications 15 processors.

N-1 may be a factor of P, where N is the maximum number of nodes in a node pencil of the array. N-1 may equal P. Alternatively, M may be a factor of P, where M is the node array dimension. P may equal M, the node array dimension. Further, both N-1 and M may be a factor of P.

20 Figure 5 depicts a two-dimensional plex communication grid with N=4 nodes **500**, each node **500** containing six ports, two communications processors, each coupled to three ports in accordance with certain embodiments of the invention.

At least some of the nodes **500** may comprise multiple coupled communications processors, also known herein as Communications Processing Units (CPU).

Each CPU contains up to N-1 ports coupled to a communication pencil in one orthogonal direction. Differing CPU's may contain differing numbers of ports, 5 indicating differing numbers of nodes **500** in node pencils of differing orthogonal directions or additional communication to external networks, mass storage, database engines or servers, or other functional components.

M may be two. Such embodiments of the invention advantageously support two-dimensional plex communications networks. Such embodiments of the invention 10 provide communication between any two nodes in at most two hops, which is the same performance of communication as between nodes through a switch, but at considerably lower levels of complexity in terms of the interconnect scheme.

N may be four. Such embodiments of the invention advantageously support two-dimensional plex communications networks with $16=4^2$ nodes. Such 15 embodiments of the invention have communication between any two nodes in at most two hops, compared to a toroidal 2-D mesh scheme, which requires up to four hops. A total direct-interconnect scheme would require 15 ports on each node, compared with six for this embodiment of the invention. A 16 port communications switch is significantly more complex than the communication 20 pencils of the plex communication grid taken collectively.

Figure 6 depicts the two communications pencils **410** and **320** coupled to Node 2,1 (**500**) and the two node pencils of Node 2,1 of Figure 4D in accordance with certain embodiments of the invention.

The node pencil in the first orthogonal direction includes Node 0,1, Node 1,1,

5 Node 2,1 and Node 3,1. The couplings of these nodes to communication pencil **410** provides a direct interconnect between each node of the node pencil to the other nodes of the node pencil.

The node pencil in the second orthogonal direction includes Node 2,0, Node 2,1,

Node 2,2 and Node 2,3. The couplings of these nodes to communication pencil

10 **320** provides a direct interconnect between each node of the node pencil to the other nodes of the node pencil.

Figure 7 depicts the communications pencils and their coupling to the node

15 pencils in the first orthogonal direction of the two-dimensional plex

communications grid of Figure 4D in accordance with certain embodiments of the

invention. The node pencils are shown each comprised of the vertical node

20 columns in Figure 7.

The first displayed node pencil **400** in the first orthogonal direction includes

Node(0,0), Node(1,0), Node(2,0) and Node(3,0). The couplings of these nodes

25 **500** to communication pencil **400** provides a direct interconnect between each

node **500** of the node pencil to at least the other nodes **500** of the node pencil.

The second displayed node pencil **410** in the first orthogonal direction includes

Node(0,1), Node(1,1), Node(2,1) and Node(3,1). The couplings of these nodes to

communication pencil **410** provides a direct interconnect between each node **500** of the node pencil at least to the other nodes **500** of the node pencil.

The third displayed node pencil **420** in the first orthogonal direction includes Node(0,2), Node(1,2), Node(2,2) and Node(3,2). The couplings of these nodes to communication pencil **420** provides a direct interconnect between each node **500** of the node pencil to the other nodes **500** of the node pencil.

The fourth displayed node pencil **430** in the first orthogonal direction includes Node(0,3), Node(1,3), Node(2,3) and Node(3,3). The couplings of these nodes **500** to communication pencil **430** provides a direct interconnect between each node **500** of the node pencil at least to the other nodes **500** of the node pencil.

Each of these node pencils couples to a corresponding communication pencil providing complete direct interconnect between at least pairs of nodes **500** of the node pencils. The communication pencils may include communication paths supporting the direct interconnect by use of N-1 ports at each node **500** coupling to the communication pencil to provide the complete direct interconnect.

Figure **8** depicts the communications pencils and their coupling to the node pencils in the second orthogonal direction of the two-dimensional plex communications grid of Figure **4D** in accordance with certain embodiments of the invention. The node pencils are each comprised of the horizontal node **500** rows in Figure **8**.

The first displayed node pencil **300** in the second orthogonal direction includes Node(0,0), Node(0,1), Node(0,2) and Node(0,3). The couplings of these nodes

500 to communication pencil 300 provides a direct interconnect between each node 500 of the node pencil at least to the other nodes 500 of the node pencil.

The second displayed node pencil 310 in the second orthogonal direction includes Node(1,0), Node(1,1), Node(1,2) and Node(1,3). The couplings of these 5 nodes 500 to communication pencil 310 provides a direct interconnect between each node 500 of the node pencil to at least the other nodes 500 of the node pencil.

The third displayed node pencil 320 in the second orthogonal direction includes Node(2,0), Node(2,1), Node(2,2) and Node(2,3). The couplings of these 10 nodes 500 to communication pencil 320 provides a direct interconnect between each node 500 of the node pencil to at least the other nodes 500 of the node pencil.

The fourth displayed node pencil 330 in the second orthogonal direction includes Node(3,0), Node(3,1), Node(3,2) and Node(3,3). The couplings of these 15 nodes 500 to communication pencil 330 provides a direct interconnect between each node 500 of the node pencil at least to the other nodes 500 of the node pencil.

Each of these node pencils couples to a corresponding communication pencil providing complete direct interconnect between pairs of nodes 500 of the node pencils. The communication pencils may include communication paths supporting the direct interconnect by use of N-1 ports at each node 500 coupling 20 to the communication pencil to provide the complete direct interconnect.

Figure 9 depicts the communications pencils 410 and 320 coupled to Node 2,1 highlighting communication pencil 410 in the first orthogonal direction of Figures

4D and 6. The communication paths of the communication pencil **410** in the first orthogonal direction are shown in solid lines, and the communications paths of the communication pencil in the other orthogonal direction and nodes **500** of the node pencils are shown with broken lines.

5 Figure **10** depicts the communications pencils **410** and **320** coupled to Node 2,1 highlighting communication pencil **320** in the second orthogonal direction of Figures **4D** and **6**. The communication paths of the communication pencil in the second orthogonal direction are shown in solid lines, and the communications paths of the communication pencil in the other orthogonal direction and nodes 10 **500** of the node pencils are shown with broken lines.

Figure **11** depicts the communications pencils **410** and **320** coupled to Node 2,1 highlighting the node pencil in the first orthogonal direction of Figures **4D** and **6**. The nodes **500** of the node pencil in the first orthogonal direction are shown in solid lines, and the communications paths of the communication pencils and 15 nodes **500** of the other node pencil are shown with broken lines.

Figure **12** depicts the communications pencils **410** and **320** coupled to Node 2,1 highlighting the node pencil in the second orthogonal direction of Figures **4D** and **6**. The nodes **500** of the node pencil in the second orthogonal direction are shown in solid lines, and the communications paths of the communication pencils 20 and nodes **500** of the other node pencil are shown with broken lines.

Figure **13A** depicts the difference between the embodiments of the invention depicted in Figure **4D** of a 2-D, N=4 plex communications grid and the

communications grid of a 2-D toroidal mesh interconnecting an N by N grid of nodes as shown in Figure **1C**. In Figure **13A**, communications paths common to the toroidal mesh and plex communications grid are shown with solid lines. Communications paths found only in the plex communications grid are shown in 5 broken lines.

Figure **13B** depicts a 2-D, $N=4$ toroidal grid of the prior art mapped onto a torus, in a fashion similar to Figure **2B**. Figure **13C** depicts the difference between the embodiments of the invention depicted in Figure **4D** of a 2-D, $N=4$ plex communications grid and the communications grid of a 2-D toroidal mesh 10 interconnecting an N by N grid of nodes as shown in Figure **13B**. Figure **13C** is equivalent to the connectivity of Figure **13A** and has been provided to show in a more graphic form distinctions pointed out in Figure **13A**.

Several things become apparent from study of Figure **13A**, the mesh array of Figure **1C** and Figures **13B** and **13C**. First, each node in a 2-D mesh, whether or 15 not it is toroidal, requires no more than four ports to interconnect with the communications network. Plex communications nodes require more. For the 2-D case, with $N=4$, each node of coupled to the plex communication grid as shown in Figure **4D** requires six ports. It is not possible for the mesh node with no more than 4 ports of Figure **1C** to substitute functionally for the nodes of the plex 20 communications scheme as shown in Figure **4D**.

Secondly, it takes two hops for communication between Node (2,0) and Node(2,2) in the mesh communications schemes, whether toroidal or not. In the

plex communications grid, it takes one hop to communicate between Node(2,0) and Node(2,2), or any other pair of nodes of that row.

Thirdly, it takes two hops for communication from Node(2,2) to Node(0,2) in the

mesh communications schemes, whether toroidal or not. In the plex

5 communications grid, it takes one hop to communicate between these nodes, or

any other pair of nodes of that column.

Thus, it can take four hops for a communication to go from Node(2,0) to

Node(0,2) in the above described mesh communications schemes, whereas it

takes only two hops in the plex communications grid. Note when N=6, a two-

10 dimensional toroidal mesh communications scheme can take up to six hops to

communicate between Node(3,0) and Node(0,3), whereas the plex

communications grid would still require at most two hops to communicate

between any two nodes.

M may be three. Such embodiments of the invention advantageously support

15 three-dimensional plex communications networks.

Figure 14 depicts a three-dimensional array of N=5 nodes **500** with orthogonal

directions **490**, **492** and **494** in accordance with the invention. In Figure 14, each

intersection of lines depicts a node **500**. This has been shown schematically to

simplify Figure 14, and is not meant to limit the contents of a node in any way.

20 Figure 15 depicts some of the node pencils and corresponding communications

pencils in the first orthogonal direction of a three-dimensional array of 4*5*5

nodes **500** in accordance with certain embodiments of the invention. The first

node pencil shown in the first orthogonal direction contains Node(0,0,2), Node(1,0,2), Node(2,0,2), and Node(3,0,2). The first communication pencil **700** is coupled to the first node pencil providing direct interconnection between each pair of nodes **500** of the first communication pencil.

5 The second node pencil in the first orthogonal direction contains Node(0,1,2), Node(1,1,2), Node(2,1,2), and Node(3,1,2). The second communication pencil **702** is coupled to the second node pencil providing direct interconnection between each pair of nodes **500** of the second communication pencil.

10 The third node pencil in the first orthogonal direction contains Node(0,2,2), Node(1,2,2), Node(2,2,2), and Node(3,2,2). The third communication pencil **704** is coupled to the third node pencil providing direct interconnection between each pair of nodes **500** of the third communication pencil.

15 The fourth node pencil in the first orthogonal direction contains Node(0,3,2), Node(1,3,2), Node(2,3,2), and Node(3,3,2). The fourth communication pencil **706** is coupled to the fourth node pencil providing direct interconnection between each pair of nodes **500** of the fourth communication pencil.

20 The fifth node pencil in the first orthogonal direction contains Node(0,4,2), Node(1,4,2), Node(2,4,2), and Node(3,4,2). The fifth communication pencil **708** is coupled to the fifth node pencil providing direct interconnection between each pair of nodes **500** of the fifth communication pencil.

Each node pencil in the first orthogonal direction contains four nodes. In this embodiment of the invention, each node may contain three ports coupled to the corresponding communication pencil to provide complete direct interconnect.

Figure 16 depicts some of the node pencils and corresponding communications pencils in the second orthogonal direction of a three-dimensional array of 4*5*5 nodes 500 in accordance with certain embodiments of the invention.

The first shown node pencil contains Node(0,0,2), Node(0,1,2), Node(0,2,2), Node(0,3,2), and Node(0,4,2). The first communication pencil 730 is coupled to the first node pencil providing direct interconnection between each pair of nodes 10 500 of the first communication pencil.

The second node pencil contains Node(1,0,2), Node(1,1,2), Node(1,2,2), Node(1,3,2), and Node(1,4,2). The second communication pencil 732 is coupled to the second node pencil providing direct interconnection between each pair of nodes 500 of the second communication pencil.

15 The third node pencil contains Node(2,0,2), Node(2,1,2), Node(2,2,2), Node(2,3,2), Node(2,3,2), and Node(2,4,2). The third communication pencil 734 is coupled to the third node pencil providing direct interconnection between each pair of nodes 500 of the third communication pencil.

20 The fourth node pencil contains Node(3,0,2), Node(3,1,2), Node(3,2,2), Node(3,3,2), Node(3,3,2), and Node(3,4,2). The fourth communication pencil 736 is coupled to the fourth node pencil providing direct interconnection between each pair of nodes 500 of the fourth communication pencil.

Each node pencil in the second orthogonal direction contains five nodes. In this embodiment of the invention, each node may contain four ports coupled to the corresponding communication pencil to provide complete direct interconnect.

Figure 17 depicts some of the node pencils and corresponding communications

5 pencils in the third orthogonal direction of a three-dimensional array of 4*5*5 nodes 500 in accordance with certain embodiments of the invention. The first shown node pencil contains Node(0,0,0), Node(0,0,1), Node(0,0,2), Node(0,0,3), and Node(0,0,4). The first communication pencil 750 is coupled to the first node pencil providing direct interconnection between each pair of nodes 500 of the first 10 communication pencil.

The second node pencil contains Node(0,1,0), Node(0,1,1), Node(0,1,2), Node(0,1,3), and Node(0,1,4). The second communication pencil 752 is coupled to the second node pencil providing direct interconnection between each pair of nodes 500 of the second communication pencil.

15 The third node pencil contains Node(0,2,0), Node(0,2,1), Node(0,2,2), Node(0,2,3), and Node(0,2,4). The third communication pencil 754 is coupled to the third node pencil providing direct interconnection between each pair of nodes 500 of the third communication pencil.

20 The fourth node pencil contains Node(0,3,0), Node(0,3,1), Node(0,3,2), Node(0,3,3), and Node(0,3,4). The fourth communication pencil 756 is coupled to the fourth node pencil providing direct interconnection between each pair of nodes 500 of the fourth communication pencil.

The fifth node pencil contains Node(0,4,0), Node(0,4,1), Node(0,4,2), Node(0,4,3), and Node(0,4,4). The fifth communication pencil **758** is coupled to the fifth node pencil providing direct interconnection between each pair of nodes **500** of the fifth communication pencil.

5 Each node pencil in the third orthogonal direction contains five nodes. In this embodiment of the invention, each node may contain four ports coupled to the corresponding communication pencil to provide complete direct interconnect.

Note that M may be four. Such embodiments of the invention advantageously support four-dimensional plex communications networks.

10 Figure **18A** depicts a node **500** with up to $M^*(N-1)$ ports **506** which couple to the communication pencils of the node **500** which intersect at the node **500** in an M-dimensional array in accordance with certain embodiments of the invention. A node **500** may contain $M^*(N-1)$ ports **506** which couple to the communication pencils of the node which intersect at the node in an M-dimensional array.

15 The physical transport layers of the communications paths coupled to ports **506** may be essentially the same. The communications protocols of the communications paths coupled to ports **506** may be essentially the same. In other further embodiments of the invention, the communications protocols of the communications paths coupled to ports **506** are not essentially the same.

20 The physical transport layer of the communication paths coupled to ports **506** may not be all essentially the same. In further embodiments of the invention, the communications protocols of the communications paths coupled to ports **506** are

essentially the same. In other further embodiments of the invention, the communications protocols of the communications paths coupled to ports **506** are not essentially the same.

One or more additional ports **508** may be contained in at least one node **500**.

5 The physical transport layers of communications paths coupled to two or more of these additional ports **508** may be essentially the same. The communications protocols of the communications paths coupled to ports **508** may be essentially the same. In other further embodiments of the invention, the communications protocols of the communications paths coupled to ports **508** are not essentially 10 the same.

The physical transport layer of communications paths coupled to two or more of these additional ports **508** may not be essentially the same. In further embodiments of the invention, the communications protocols of the communications paths coupled to ports **508** are essentially the same. In other 15 further embodiments of the invention, the communications protocols of the communications paths coupled to ports **508** are not essentially the same.

At least one node **500** may be accessibly coupled **502** to memory **504**. In further embodiments of the invention, node **500** contains an instruction processor further accessibly coupled **502** to memory **504**. In further embodiments of the invention, 20 node **500** contains at least two instruction processors. In further embodiments of the invention, node **500** contains multiple instruction processors accessibly coupled **502** to memory **504**.

Recall that each node may include a communication processor. Each node may further include P communications processors. P may be a factor of the number of communications ports required by the communications grid to couple with the node. The number of required ports may be $M^*(N-1)$, so that P becomes a factor 5 of $M^*(N-1)$. Such embodiments of the invention advantageously support communications processing at the node partitioned across the P communications processors.

Further recall that N-1 may be a factor of P, where N is the maximum number of 10 nodes in a node pencil of the array. N-1 may equal P. Alternatively, M may be a factor of P, where M is the node array dimension. P may equal M, the node array dimension. Further, both N-1 and M may be a factor of P.

The P communications processors may be coupled by a bus. Such embodiments of the invention advantageously support the use of a bus to couple the P communications processors of a node. Further additional embodiments of the 15 invention include the bus coupling the P communications processors supporting a bus master. Further embodiments of the invention include the bus master as one of the P communications processors. Further embodiments of the invention include the bus master, over time, being any of the P communications processors.

20 As used herein, a bus refers to a common communication coupling between multiple communicating devices. As used herein, bus master refers to a device controlling which of the communicating devices coupled to the controlled bus

may actively communicate or access the bus. It is common for some of the coupled communicating devices to have to wait for bus access.

Figure 18B depicts a node 500 with up to P CPU's 510, 520, 550 and 560 coupled by 530-536 to the communication pencils of node 500 which intersect at 5 node 500 in an M-dimensional array in accordance with certain embodiments of the invention. To minimize the complexity of the drawing and discussion, Figure 18B shows embodiments of the invention for P equal to four. This is done strictly to minimize the complexity of the discussion and not to impose a limitation upon interpretation of the claims. In certain embodiments of the invention, P is two. In 10 other further embodiments of the invention, P is three. In other further embodiments of the invention, P is four. In certain other embodiments of the invention, P is at least five.

At least one node 500 contains P CPU's 510, 520, 550 and 560 coupled by 530-536. Couplings 530-536 may form a single shared coupling 530. A bus may 15 provide coupling 530. A bus may be considered as a resource allowing a limited subset of processors to simultaneously communicate. Buses are often found to require at least some combinations of processors to sometimes wait before communicating, which is often referred to as bus access.

A bus arbitration scheme may control access to coupling 530. A bus master may 20 further control the bus providing coupling 530. In further embodiments of the invention, one of the CPUs acts as the bus master controlling coupling 530. In further embodiments of the invention, any of the CPUs may act as the bus

master controlling coupling **530**. In further embodiments of the invention, all of the CPUs occasionally act as the bus master controlling coupling **530**.

Couplings **530-536** may form a single shared coupling **530** of the CPUs **510** and **520** with a specific interface **532** via **534** to CPU **550** and via **536** to CPU **550**.

5 Couplings **534** and **536** may further act as a bus with interface **532** acting as a bridge between coupling **530** and bus **534-536**.

Couplings **530-536** may form a single shared coupling **530** of the CPUs **510** and **520** with coupling **530-532-534** acting as a direct interface of CPU **550** to one of CPUs **510** and **520**. Coupling **530-532-536** may further act as a direct interface of 10 CPU **560** to one of CPUs **510** and **520**. Coupling **530-532-536** may also further act as a direct interface of CPU **560** to the shared coupling **530** of CPUs **510** and **520**.

Node **500** may contain CPU **510** further accessibly coupled **512** to memory **514**. Node **500** may contain CPU **520** accessibly coupled **522** to memory **524**. Node 15 **500** may contain CPU **550** accessibly coupled **552** to memory **554**. Node **500** may contain CPU **560** accessibly coupled **562** to memory **564**.

Two or more of the memories **514**, **524**, **554** and **564** may be contained within a single package. As used herein, a package includes but is not limited to a printed circuit board, an assembly including a printed circuit board, a multi-chip module, 20 an integrated circuit, a circuit encased in one or more of the collection, including but not limited to, plastic, ceramic, metallic and optically conductive materials.

The package may further include power distribution components. The package may also further include thermal dissipation components.

One or more of the CPUs **510, 520, 550, and 560** may be contained within a single package with the respective accessibly coupled memories **514, 524, 554** and **564**.

5

At least one of the CPUs **510** may contain up to $M^*(N-1)$ ports **516**. Each of the CPUs **510, 520, 550, and 560** may contain up to $M^*(N-1)/P$ ports **516**. Each of the CPUs **510, 520, 550, and 560** may further contain at least $N-1$ ports **516, 526, 556** and **566**, coupling to the communication paths of communication pencils in the M orthogonal directions. In further embodiments of the invention, each of the CPUs **510, 520, 550, and 560** contain $N-1$ ports **516, 526, 556**, and **566** coupling to the communication paths of communication pencils in the M orthogonal directions. Each of the CPUs **510, 520, 550, and 560** may contain M ports **516, 526, 556** and **566**, coupling to the communication paths of communication pencils in the M orthogonal directions.

10
15

At least one of the CPUs **510** may contain at least one additional port **518**. In further embodiments of the invention, each of the CPUs **510, 520, 550** and **560** may contain at least one additional port **518, 528, 558** and **568**. In certain embodiments of the invention one or more of these additional ports couple to external communications networks. In certain embodiments of the invention one or more of these additional ports couple to an external database engine. In certain embodiments of the invention one or more of these additional ports couple to a server.

20

At least one of the communications processors may be further comprised of at least one instruction processor accessibly coupled to a memory. Such embodiments of the invention advantageously support instruction processing at each of the communications processors.

- 5 At least one of the communications processors **510** may include a communication handler coupled to at least one of the ports **516**. The communication handler may include, but is not limited to, a finite state machine. The finite state machine may include, but is not limited to, one or more programmable logic circuits, Field Programmable Logic Arrays (FPGAs), gate arrays, standard cell circuits. Such embodiments of the invention advantageously support protocol handlers for digital protocols, which are often advantageously implemented, at least in part, as finite state machines. Note that the finite state machine may change state synchronously or asynchronously. Synchronous finite state machines may use a synchronizing mechanism based
- 10 15 upon the condition of coupled port(s) **516**, or based upon one or more conditions within the node **500**.

The communication handler may further include, but is not limited to, transmitter, receiver or transceiver circuitry interfacing to at least one of the physical transport layers of the port, for at least one of the ports. Such embodiments of the invention advantageously support physical transport layer interfaces.

At least one, possibly all, of the communications processors may be comprised of a communications instruction processor accessibly coupled to the memory. The communications instruction processor is communicatively coupled to at least

one of the ports. The communications processor may be communicatively coupled to the port via the communication handler. Such embodiments of the invention advantageously support communications instruction processors coupled to at least some of the ports. Programmable communications processing 5 further advantageously supports encryption, security, and other activities requiring reconfiguration over time.

The M communications processors may be coupled by a direct connection network of each of the M communications processors coupled directly to each of the remaining communications processors. Such embodiments of the invention 10 advantageously support the communications processors coupled by a direct connection network, where each communications processor is directly coupled to every other communications processor of the node. This advantageously avoids having to wait for bus access.

Figure 19 depicts a node 500 with up to P CPU's 510, 520, 550 and 560 directly 15 coupled by 538-548, with ports 516, 526, 556 and 566 which couple to the communication pencils of the node pencils intersecting at the node 500 in an M-dimensional array in accordance with certain embodiments of the invention. To minimize the complexity of the drawing and discussion, Figure 19 shows 20 embodiments of the invention for P equal to four. This is done strictly to minimize the complexity of the discussion and not to impose a limitation upon interpretation of the claims. In further embodiments of the invention, P is two. In other further embodiments of the invention, P is three. In other further

embodiments of the invention, P is four. In certain other embodiments of the invention, P is at least five.

As in Figure 18B, at least one node 500 contains P coupled CPU's 510, 520, 550 and 560. Node 500 contains CPU 510 further accessibly coupled 512 to memory

5 514. In further embodiments of the invention, node 500 contains CPU 520 accessibly coupled 522 to memory 524. In further embodiments of the invention, node 500 contains CPU 550 accessibly coupled 552 to memory 554. In further embodiments of the invention, node 500 contains CPU 560 accessibly coupled 562 to memory 564.

10 As in Figure 18B, two or more of the memories 514, 524, 554 and 564 may be contained within a single package.

As in Figure 18B, one or more of the CPUs 510, 520, 550, and 560 may be contained within a single package with the respective accessibly coupled memories 514, 524, 554 and 564.

15 As in Figure 18B, at least one of the CPUs 510 may contain up to $M^*(N-1)$ ports 516. Each of the CPUs 510, 520, 550, and 560 may contain up to $M^*(N-1)/P$ ports 516. Each of the CPUs 510, 520, 550, and 560 may further contain at least N-1 ports 516, 526, 556 and 566, coupling to the communication paths of communication pencils in the M orthogonal directions. In further embodiments of 20 the invention, each of the CPUs 510, 520, 550, and 560 contain N-1 ports 516, 526, 556, and 566 coupling to the communication paths of communication pencils in the M orthogonal directions. Each of the CPUs 510, 520, 550, and 560

may contain M ports **516, 526, 556** and **566**, coupling to the communication paths of communication pencils in the M orthogonal directions.

As in Figure **18B**, at least one of the CPUs **510** may contain at least one additional port **518**. In further embodiments of the invention, each of the CPUs

5 **510, 520, 550** and **560** contain at least one additional port **518, 528, 558** and **568**. In certain embodiments of the invention one or more of these additional ports couple to external communications networks. In certain embodiments of the invention one or more of these additional ports couple to an external database engine. In certain embodiments of the invention one or more of these additional 10 ports couple to a server.

When $P=2$, CPU **510** couples via **538** to CPU **520**, which is similar to the situation described in Figure **18B**. When $P=3$, additional couplings **540** and **542** connect CPU **510** with CPU **550** and CPU **520** with CPU **550**, respectively. This supports each CPU being able to independently directly communicate with any of 15 the other CPUs without having to wait for bus access.

When $P=4$, additional couplings **548, 546** and **544** connect CPU **510** with CPU **560**, CPU **520** with CPU **560** and CPU **550** with CPU **560**, respectively. This supports each CPU being able to independently directly communicate with any of the other CPUs without having to wait for bus access.

20 Note that the physical transport layers of the couplings **538-548** may differ from the physical transport layers of one or more of the communication pencils coupled to the various ports **516, 518, 526, 528, 566, 568, 556**, and **558**. The

physical transport layers of the couplings **538-548** of one node **500** may differ from another node's physical transport layers for couplings **538-548**.

Certain nodes may be implemented as in Figure **18B**, while other nodes may be implemented as in Figure **19**. Certain plex communications grids may use at 5 least one node, which is itself a plex communication grid.

Node level handling of communications processing and routing is a well developed topic in the prior art which one of ordinary skill readily understands to include, but not be limited to, message passing, stream processing, encryption, error control coding, gateways, fire walls, TCP-IP, Internet, and web-sites.

10 Communication across a communication pencil includes traversal of the physical transport(s) of the relevant communication path of the communication pencil. Thus, traversing the communication grid along a collection of communication pencils includes traversal of the physical transports of communication paths of those communication pencils.

15 As is obvious to one of ordinary skill, the physical transport layers of the communication paths within a node as well as communication paths within a communication pencil include, but are not limited to, one or more wires, twisted pairs of wires, and wave guides. Wave guides as used herein include, but are not limited to, coaxial cable, fiber optics and micro-channels. These physical 20 transports may further support communications protocols in the radio, microwave, infra-red and optical, ultra-violet frequency domains. Such protocols include but are not limited to frequency modulation, time division multiple access,

wavelet division multiple access, wavelength division multiple access (WDM) and soliton transmission technologies.

As is obvious to one of ordinary skill, traversing a physical transport layer may include entering the physical transport layer, crossing the physical transport layer and leaving the physical transport layer. Entering and leaving a physical transport layer may be performed by, at least, various electronic, electro-optical, opto-electronic and resonant cavity devices including but not limited to diode and transistor structures, lasers and various tuned crystalline structures.

As is obvious to one of ordinary skill, traversing a first and a second physical transport layer may include the following. Traversing the first physical transport layer. Traversing between the first physical transport layer and the second physical transport layer. And traversing the second physical transport layer. Traversing between first and second physical transport layers may be performed by, at least, various electronic, electro-optical, opto-electronic and resonant cavity devices including but not limited to diode and transistor structures, lasers and various tuned crystalline structures.

Certain embodiments of the invention include a method of communicating from a first node to a second node in a multi-dimensional lattice of nodes through a communications network, in accordance with certain embodiments of the invention. The second node differs from the first node in R dimensional components, where R is a number between 0 (if the nodes are the same) and M (if the nodes differ in every orthogonal direction's component). The communication traverses a node pencil path comprised of R-1 successive

intermediate nodes. Each of the successive intermediate nodes has one less dimensional component differing from the second node. Each node pencil has a coupled corresponding communication pencil containing at least one physical transport.

5 Figure 20A depicts a system 600 containing an M=2, N=4 plex communication grid with $16=N^M$ nodes 500, with one node 500 including four communication processing units (CPUs), three nodes 500 including three CPUs, and 12 nodes 500 including two CPUs, where the 3 CPU nodes each are further coupled to external communications networks and each possess a tunneling interface, in
10 accordance with certain embodiments of the invention.

Nodes 500 (0,0), (1,1), and (2,2) each include three communication processing units (CPUs) 510, 520, and 550. Node 500 (3,3) includes four CPUs 510, 520, 550, and 560. All Nodes 500 include CPUs 510 and 520. Each CPU 510 is accessibly coupled 512 to memory 514. Each CPU 520 is accessibly coupled 522 to memory 524. Each CPU 550 is accessibly coupled 552 to memory 554.
15 Note that the reference numbers 512, 522 and 552 are not shown to minimize the complexity of the figure. CPU 560 may further include an accessibly coupled memory.

Each of the CPUs 550 includes a tunneling interface coupling to a
20 communications tunnel. CPU 550 of Node 500 (0,0) couples to communications tunnel 640. CPU 550 of Node 500 (1,1) couples to communications tunnel 642. CPU 550 of Node 500 (2,2) couples to communications tunnel 644. CPU 550 of Node 500 (3,3) couples to communications tunnel 646.

Any node **500** may include additional communications interfaces to add further communications capabilities.

By way of example, Figure **20A** shows Node **500** (1,1) CPU **550** may include communications interfaces **610** and **612**. Node **500** (2,2) CPU **550** may include 5 communications interfaces **606** and **608**. Node **500** (3,3) CPU **550** may include communications interfaces **602** and **604**. One or more of these CPU **550** may include circuitry to select one of the two communications interfaces to be actively utilized. One or more of these CPU **550** may concurrently use both communications interfaces.

10 Figure **20A** shows Node **500** (3,3) CPU **520** with an accessibly coupled memory **524** which includes a ROM.

Any node **500** may have one or more CPUs accessibly coupled to one or more memories. That accessibly coupled memory may include an accessibly coupled 15 non-volatile memory component. That accessibly coupled non-volatile memory component may include ROM, flash memory, EPROM, EEPROM, CD-ROM, DVD-ROM components accessibly coupled to one or more of the CPUs of a node. The non-volatile memory component may further support a file management system interface through the accessibly coupled CPU.

Figure **20A** shows Node **500** (3,3) CPU **560** has a communication interface **670** 20 which may further couple to a mass storage system, database engine, network gateway, or specialized engine.

Any node **500** may include such additional communication capabilities. The system **600** may further support distributed access to such resources by use of an access protocol which is compatible with the communications protocols in use on at least some of the communication pencils. By way of example, file access 5 to a mass storage system may be through a TCP-IP compatible protocol, such as LDAP. Access to a database engine may be through a TCP-IP compatible protocol such as XML or through a CORBA compatible object structure protocol.

Figure **20B** alternatively depicts a system **600** containing an $M=2$, $N=4$ plex communication grid with $16=N^M$ nodes **500**, with one node **500** including four 10 communication processing units (CPUs), three nodes **500** including three CPUs, and 12 nodes **500** including two CPUs, where the 3 CPU nodes each are further coupled to external communications networks and each possess a tunneling interface, in accordance with certain embodiments of the invention.

As in Figure **20A**, Nodes **500** (1,1), and (2,2) each include three communication 15 processing units (CPUs) **510**, **520**, and **550**.

Node **500** (0,0) includes four CPUs **510**, **520**, **550**, and **560**. Nodes **500** (3,3) includes three communication processing units (CPUs) **510**, **520**, and **550**.

As in Figure **20A**, all Nodes **500** include CPUs **510** and **520**. Each CPU **510** is accessibly coupled **512** to memory **514**. Each CPU **520** is accessibly coupled 20 **522** to memory **524**. Each CPU **550** is accessibly coupled **552** to memory **554**. Note that the reference numbers **512**, **522** and **552** are not shown to minimize

the complexity of the figure. CPU 560 may further include an accessibly coupled memory.

As in Figure 20A, each of the CPUs 550 includes a tunneling interface coupling to a communications tunnel.

- 5 However, Figure 20B shows CPU 550 of Node 500 (3,3) couples to communications tunnel 640. CPU 550 of Node 500 (2,2) couples to communications tunnel 642. CPU 550 of Node 500 (1,1) couples to communications tunnel 644. CPU 550 of Node 500 (0,0) couples to communications tunnel 646.
- 10 As in Figure 20A, any node 500 may include additional communications interfaces to add further communications capabilities.

Figure 20B shows Node 500 (3,3) CPU 550 may include communications interfaces 610 and 612. Node 500 (2,2) CPU 550 may include communications interfaces 606 and 608. Node 500 (1,1) CPU 550 may include communications

- 15 interfaces 602 and 604. One or more of these CPU 550 may include circuitry to select one of the two communications interfaces to be actively utilized. One or more of these CPU 550 may concurrently use both communications interfaces.

Figure 20B shows Node 500 (0,0) CPU 520 with an accessibly coupled memory 524 which includes a ROM.

- 20 Any node 500 may have one or more CPUs accessibly coupled to one or more memories. That accessibly coupled memory may include an accessibly coupled

non-volatile memory component. That accessibly coupled non-volatile memory component may include ROM, flash memory, EPROM, EEPROM, CD-ROM, DVD-ROM components accessibly coupled to one or more of the CPUs of a node. The non-volatile memory component may further support a file management system interface through the accessibly coupled CPU.

Figure 20B shows Node 500 (0,0) CPU 560 has a communication interface 670 which may further couple to mass storage system, database engine, network gateway, or specialized engine.

Any node 500 may include such additional communication capabilities. The system 600 may further support distributed access to such resources by use of an access protocol which is compatible with the communications protocols in use on at least some of the communication pencils. By way of example, file access to a mass storage system may be through a TCP-IP compatible protocol, such as LDAP. Access to a database engine may be through a TCP-IP compatible protocol such as XML or through a CORBA compatible object structure protocol.

Figure 21 depicts a system 800 including two instances of system 600 as depicted in Figures 20A and 20B, referred to as 600-1 and 600-2, in accordance with certain embodiments of the invention.

Figure 21 depicts system 800 including 6 external communications mechanisms 802-812 each coupling respectively to 602-612 of system 600-1 and coupling respectively to 602-612 of system 600-2. Note that these external communication mechanisms may use the same physical transport layer

mechanism, or that they may differ. Whether or not these external communications mechanisms use the same physical transport layer mechanism, they may use similar communications protocols, or they may use dissimilar communications protocols.

5 Figure 21 depicts system 800 including resource pool 830 coupled 832 to system 600-1 by communication interface 670. The resource pool 830 is coupled 834 to system 600-2 by communication interface 670. The resource pool 830 may include but is not limited to one or more of the following: mass storage systems, database engines, network gateways and specialized engines. Couplings 832 and 834 may each include more than one communication mechanism. Each of the communications mechanisms included in couplings 832 and 834 may involve more than one physical transport layer. The communication mechanisms included in couplings 832 and 834 may or may not employ similar physical transport layers. The communication mechanisms included in couplings 832 and 834 may or may not employ compatible communications protocols.

10

15

The systems 600-1 and 600-2 may further support distributed access to Resource Pool 830 by use of an access protocol compatible with communications protocols in use on at least some of the communication pencils of their respective internal communications grids. By way of example, file access to a mass storage system may be through a TCP-IP compatible protocol, such as 20 LDAP. Access to a database engine may be through a TCP-IP compatible protocol such as XML or through a CORBA compatible object structure protocol.

System **600-1** may be provided **842** power supply **840**. System **600-2** may be provided **846** power supply **844**. Such independent power supplies can cost-effectively add to the overall reliability of the system **800**, because if one power supply fails, at least half of the system **600** components continue to function.

5 The total cost of two of these power supplies is considerably less than the cost a single power supply, plus supply backup to power both system **600**'s.

Certain embodiments of the invention are applicable to other multi-dimensional arrays of computers.

Figure **22** partially depicts a toroidal three-dimensional mesh communication grid

10 for an $N=3$, three dimensional ($M=3$) array of nodes **500**, each comprising $P=2$ Communication Processor Units (CPU) which each comprise $M=3$ ports for the corresponding communication pencils.

Consider communication between a first node **500** and a second node **500**, when

the first node and second node are both coupled to communication pencils

15 coupling to a third node **500**. The first node **500** couples to a first communication pencil coupled to a third node **500**. The third node **500** couples to a second communication pencil coupled to the second node **500**. Each of the communication pencils includes at least one physical transport layer.

Figure **23A** depicts a flowchart of a method communicating between a first node

20 **500** and a second node **500** when the first node **500** and second node **500** are both coupled to communication pencils coupling to a third node **500**, in accordance with certain embodiments of the invention.

Operation **1000** starts the operations of this flowchart. Arrow **1002** directs the flow of execution between operation **1000** and operation **1004**. Operation **1004** performs the first node communicating with the third node via the first communication pencil. Arrow **1006** directs execution between operation **1004** and operation **1008**. Operation **1008** performs the third node communicating with the second node via the second communication pencil. Arrow **1010** directs execution between operation **1008** and operation **1012**. Operation **1012** terminates the operations of this flowchart.

Note that Figure **23A** permits the operations of this flowchart to begin at the exit operation **1012** and exit at the beginning operation **1000**. This, while unusual, has been done to point out the ability of any element of the communication being an initiator or terminator of the operations depicted in this Figure.

Figure **23B** depicts a detail flowchart of operation **1004** of Figure **23A** further performing the first node communicating with the third node via the first communication pencil.

Arrow **1030** directs the flow of execution from starting operation **1004** to operation **1032**. Operation **1032** performs traversing all of the physical transport layers included in the first communication pencil. Arrow **1034** directs execution from operation **1032** to operation **1036**. Operation **1036** terminates the operations of this flowchart.

Figure 23C depicts a detail flowchart of operation 1012 of Figure 23A further performing the third node communicating with the second node via the second communication pencil.

Arrow 1050 directs the flow of execution from starting operation 1012 to 5 operation 1052. Operation 1052 performs traversing all of the physical transport layers included in the second communication pencil. Arrow 1054 directs execution from operation 1052 to operation 1056. Operation 1056 terminates the operations of this flowchart.

In certain embodiments of the invention, a communication pencil may include 10 exactly one physical transport layer. However, a communication pencil may include a first physical transport layer and a second physical transport layer. Two communication pencils of a single embodiment of the invention may include different physical transport layers and differing numbers of physical transport layers. A communication pencil may include a bus. The communication pencil 15 may implement a Time Division Multiple Access (TDMA) protocol.

Figure 24A depicts a detail flowchart of operation 1032 of Figure 23B further performing traversing the physical transport layers when the first communication pencil includes a first physical transport layer and second physical transport layer.

20 Arrow 1070 directs the flow of execution from starting operation 1032 to operation 1072. Operation 1072 performs traversing the first physical transport

layer. Arrow **1074** directs execution from operation **1072** to operation **1076**.

Operation **1076** terminates the operations of this flowchart.

Arrow **1080** directs the flow of execution from starting operation **1032** to operation **1082**. Operation **1082** performs traversing between the first physical

5 transport layer and the second physical transport layer. Arrow **1084** directs execution from operation **1082** to operation **1076**. Operation **1076** terminates the operations of this flowchart.

Arrow **1090** directs the flow of execution from starting operation **1032** to operation **1092**. Operation **1092** performs traversing the second physical

10 transport layer. Arrow **1094** directs execution from operation **1092** to operation **1076**. Operation **1076** terminates the operations of this flowchart.

Figure **24B** depicts a detail flowchart of operation **1052** of Figure **24C** further performing traversing the physical transport layers when the communication pencil includes a first physical transport layer and second physical transport

15 layer.

Arrow **1110** directs the flow of execution from starting operation **1052** to operation **1112**. Operation **1112** performs traversing the first physical transport

layer. Arrow **1114** directs execution from operation **1112** to operation **1116**.

Operation **1116** terminates the operations of this flowchart.

20 Arrow **1120** directs the flow of execution from starting operation **1052** to operation **1122**. Operation **1122** performs traversing between the first physical transport layer and the second physical transport layer. Arrow **1124** directs

execution from operation **1122** to operation **1116**. Operation **1116** terminates the operations of this flowchart.

Arrow **1130** directs the flow of execution from starting operation **1052** to operation **1132**. Operation **1132** performs traversing the second physical transport layer. Arrow **1134** directs execution from operation **1132** to operation **1116**. Operation **1116** terminates the operations of this flowchart.

By way of example of the operation of Figures **23A-23C**, **24A** and **24B**, consider communication for a 2-D array based upon Figure **4A**. M is 2. Given any two nodes, they either have the same address in the array, differ by one dimensional component or differ by both dimensional components.

When two nodes have the same address, they are the same node. Communications within a node, while important, will be postponed briefly.

Consider two nodes differing by one dimensional component, say Node **500** (0,0) and Node **500** (1,0), which differ in the first dimensional component. Node **500** (0,0) couples **402** to communication pencil **400** and Node **500** (1,0) couples **404** to communication pencil **400**. Communication between Node **500** (0,0) and Node **500** (1,0) entails Node **500** (0,0) communicating with Node **500** (1,0) via communication pencil **400**. Node **500** (0,0) communicating with Node **500** (1,0) via communication pencil **400** further includes traversing all of the physical transport layers in communication pencil **400**.

Consider communication between first Node **500** (0,0) and second Node **500** (1,1), which differ by two dimensional components. These nodes are not coupled

to a common communication pencil. However, there are a number of choices of a third node sharing a common dimensional component with each of them. Such choices for the third node include Node **500** (1,0) and Node **500** (0,1). For the sake of discussion, select Node **500** (1,0). Node **500** (0,0) couples **402** to a first communication pencil **400** and Node **500** (1,0) couples **404** to first communication pencil **400**. Node **500** (1,0) couples **312** to a second communication pencil **310** and Node **500** (1,1) couples **314** to second communication pencil **310**.

Communication between first Node **500** (0,0) and second Node **500** (1,1) may be achieved by first Node **500** (0,0) communicating with third Node **500** (1,0) via first communication pencil **400** (step of **1004** of Figure **23A**) and third Node **500** (1,0) communicating with second Node **500** (1,1) via second communication pencil **310** (step of **1008** of Figure **23A**).

First Node **500** (0,0) communicating with third Node **500** (1,0) via first communication pencil **400** (step of **1004** of Figure **23A**) further includes traversing all of the physical transport layers in first communication pencil **400** (step of **1032** of Figure **23B**).

Third Node **500** (1,0) communicating with second Node **500** (1,1) via second communication pencil **310** (step of **1008** of Figure **23A**) further includes traversing all of the physical transport layers in second communication pencil **310** (step of **1052** of Figure **23C**).

A corresponding coupled communication pencil may include one of the physical transport layers. At least one of the corresponding coupled communication pencils may include a first and a second of the physical transport layers.

By way of example, in communications systems employing communications

5 pencils similar to Figure 3C of the prior art discussion, the write optical fiber from the nodes to the control point comprises the first physical transport, the traversal of the control point constitutes the traversal between the first and second physical transport layers, and traversal of the read optical fiber constitutes traversal of the second physical transport.

10 The physical transport layers of a communication pencil may include but are not limited to a wave guide. The wave guide may physically transport at least in the microwave domain. The wave guide may physically transport at least in the infrared domain. The wave guide may physically transport at least in the optical domain. The wave guide may physically transport at least in the radio domain.

15 As used herein, a wave guide may include but is not limited to at least one optical fiber, at least one coaxial cable and/or at least one wire. Further, a physical transport layer may include but is not limited to at least one twisted pair of wires.

Note that certain embodiments of the invention may include redundant communication pencils coupled to a node pencil, redundant communication

20 paths within a communication pencil, or redundant physical transport within a communication path, which may or may not involve distinct physical transport layers. The necessary physical transport layers included in a communication will

be understood to be those physical transport layers used to actually communicate between the relevant nodes.

Consider a node **500** coupling to M communication pencils, where M is at least two. The node **500** includes M communication interfaces **506**, each 5 communication interfaces coupling to a corresponding communication pencil.

In Figure 25, M will be assumed to be less than 6. This has been done for the sake of simplicity of discussion and is not meant to limit M in any way.

Figure 25 depicts a flowchart of a communication process performed within the node **500** controlling all of the M communications interfaces, where M is between 10 2 and 5, in accordance with certain embodiments of the invention.

Operation **1300** starts the operations of this flowchart. Arrow **1302** directs the flow of execution from operation **1300** to operation **1304**. Operation **1304** performs interacting within the node with the first communication interface. Arrow **1306** directs execution from operation **1304** to operation **1308**. Operation 15 **1308** terminates the operations of this flowchart.

Arrow **1310** directs the flow of execution from starting operation **1300** to operation **1312**. Operation **1312** performs interacting within the node with the second communication interface. Arrow **1314** directs execution from operation 20 **1312** to operation **1308**. Operation **1308** terminates the operations of this flowchart.

Arrow **1320** directs the flow of execution from starting operation **1300** to operation **1322**. Operation **1322** performs interacting within the node with the third communication interface. Arrow **1324** directs execution from operation **1322** to operation **1308**. Operation **1308** terminates the operations of this

5 flowchart.

Arrow **1330** directs the flow of execution from starting operation **1300** to operation **1332**. Operation **1332** performs interacting within the node with the fourth communication interface. Arrow **1334** directs execution from operation **1332** to operation **1308**. Operation **1308** terminates the operations of this

10 flowchart.

Arrow **1340** directs the flow of execution from starting operation **1300** to operation **1342**. Operation **1342** performs interacting within the node with the fifth communication interface. Arrow **1344** directs execution from operation **1342** to operation **1308**. Operation **1308** terminates the operations of this flowchart.

15 To further simplify the discussion, only operation **1304** will be expanded in subsequent flowcharts and their accompanying discussion. It is to be understood that any discussion of interaction within a node with one communication interface applies in all its variations to the interaction within a node with a different communication interface, and that two communication interfaces may differ in

20 how their interactions are embodied within a single node.

Figure 26 depicts a detail flowchart of operation 1304 of Figure 25 further performing interacting within the node with the first communication interface in accordance with certain embodiments of the invention.

Arrow 1370 directs the flow of execution from starting operation 1304 to 5 operation 1372. Operation 1372 performs receiving a first communication from the communication interface to create a received communication from the communication interface. Arrow 1374 directs execution from operation 1372 to operation 1376. Operation 1376 terminates the operations of this flowchart.

Arrow 1380 directs the flow of execution from starting operation 1304 to 10 operation 1382. Operation 1382 performs processing the received communication from the communication interface. Arrow 1384 directs execution from operation 1382 to operation 1376. Operation 1376 terminates the operations of this flowchart.

Arrow 1390 directs the flow of execution from starting operation 1304 to 15 operation 1392. Operation 1392 performs sending a local communication to the communication interface to create a second communication to the communication interface. Arrow 1394 directs execution from operation 1392 to operation 1376. Operation 1376 terminates the operations of this flowchart.

A communication processor may be included within the node 500, coupled to at 20 least one communication interface.

Figure 27A depicts a detail flowchart of operation 1372 of Figure 26 further performing for each of the communication interfaces coupled to the communication processor, receiving the first communication.

Arrow 1410 directs the flow of execution from starting operation 1372 to

5 operation 1412. Operation 1412 performs the communication processor receiving the first communication from the communication interface to create the received communication from the communication interface. Arrow 1414 directs execution from operation 1412 to operation 1416. Operation 1416 terminates the operations of this flowchart.

10 Figure 27B depicts a detail flowchart of operation 1382 of Figure 26 further performing for each of the communication interfaces coupled to the communication processor, processing the received communication.

Arrow 1430 directs the flow of execution from starting operation 1382 to

operation 1432. Operation 1432 performs the communication processor 15 processing the received communication from the communication interface.

Arrow 1434 directs execution from operation 1432 to operation 1436. Operation 1436 terminates the operations of this flowchart.

Figure 27C depicts a detail flowchart of operation 1392 of Figure 26 further performing for each of the communication interfaces coupled to the

20 communication processor, sending the local communication.

Arrow 1450 directs the flow of execution from starting operation 1392 to

operation 1452. Operation 1452 performs the communication processor sending

the local communication to the communication interface to create the second communication to the communication interface. Arrow **1454** directs execution from operation **1452** to operation **1456**. Operation **1456** terminates the operations of this flowchart.

5 Figure **28A** depicts a detail flowchart of operation **1432** of Figure **27B** further performing processing the received communication in accordance with certain embodiments of the invention.

Arrow **1510** directs the flow of execution from starting operation **1432** to operation **1512**. Operation **1512** performs determining a received communication destination based upon the received communication from the communication interface. Arrow **1514** directs execution from operation **1512** to operation **1516**. Operation **1516** terminates the operations of this flowchart.

10
15 Arrow **1520** directs the flow of execution from starting operation **1432** to operation **1522**. Operation **1522** performs routing the received communication whenever the received communication destination is external to the node. Arrow **1524** directs execution from operation **1522** to operation **1516**. Operation **1516** terminates the operations of this flowchart.

20 Arrow **1530** directs the flow of execution from starting operation **1432** to operation **1532**. Operation **1532** performs delivering the received communication whenever the received communication destination is internal to the node. Arrow **1534** directs execution from operation **1532** to operation **1516**. Operation **1516** terminates the operations of this flowchart.

Figure 28B depicts an alternative detail flowchart of operation 1432 of Figure 27B further performing processing the received communication in accordance with certain embodiments of the invention.

Arrow 1550 directs the flow of execution from starting operation 1432 to operation 1552. Operation 1552 performs determining a received communication destination based upon the received communication from the communication interface. Arrow 1554 directs execution from operation 1552 to operation 1556. Operation 1556 performs routing the received communication whenever the received communication destination is external to the node. Arrow 1558 directs execution from operation 1556 to operation 1560. Operation 1560 performs delivering the received communication whenever the received communication destination is internal to the node. Arrow 1562 directs execution from operation 1560 to operation 1564. Operation 1564 terminates the operations of this flowchart.

Note that Figures 28A and 28B depict two of several organizational approaches to implementing very similar if not the same operations. Figure 28A tends to be in keeping with many event triggered real time environments, where each of the operations may be independently and concurrently performed, possibly with queuing mechanisms applied before or after performance of one or more of these operations. Figure 28B tends to be in keeping with a determination of destination, sequentially followed by executing of one or both of the operations of routing and delivery. By way of example, flowcharts could be drawn showing only one choice, to route or to deliver, which would be in accordance with certain

other embodiments of the invention. However, for the sake of simplicity of discussion, only Figures **28A** and **28B** have been provided.

In the following discussion, only operation **1512** will be referenced for determining based upon the received communication. This is not intended to limit the scope of invention in any way, but to simplify the discussion.

Figure **29A** depicts a detail flowchart of operation **1512** of Figure **28A** further performing determining the received communication destination.

Arrow **1570** directs the flow of execution from starting operation **1512** to operation **1572**. Operation **1572** performs extracting from the received communication to create a destination component. Arrow **1574** directs execution from operation **1572** to operation **1576**. Operation **1576** terminates the operations of this flowchart.

Arrow **1580** directs the flow of execution from starting operation **1512** to operation **1582**. Operation **1582** performs evaluating the destination component to create the received communication destination. Arrow **1584** directs execution from operation **1582** to operation **1576**. Operation **1576** terminates the operations of this flowchart.

Figure **29B** depicts an alternative detail flowchart of operation **1512** of Figure **28A** further performing determining the received communication destination in accordance with certain embodiments of the invention.

Arrow **1590** directs the flow of execution from starting operation **1512** to operation **1592**. Operation **1592** performs extracting from the received communication to create a destination component. Arrow **1594** directs execution from operation **1592** to operation **1596**. Operation **1596** performs evaluating the destination component to create the received communication destination. Arrow **1598** directs execution from operation **1596** to operation **1600**. Operation **1600** terminates the operations of this flowchart.

Note that Figures **29A** and **29B** depict two of several organizational approaches to implementing very similar if not the same operations. Figure **29A** tends to be in keeping with many event triggered real time environments, where each of the operations may be independently and concurrently performed, possibly with queuing mechanisms applied before or after performance of one or more of these operations. Figure **29B** tends to be in keeping with a determination of destination, sequentially followed by executing of one or both of the operations of routing and delivery.

In the following discussion, only operation **1572** will be referenced for extracting from the received communication to create a destination component. This is not intended to limit the scope of invention in any way, but to simplify the discussion.

In the following discussion, only operation **1582** will be referenced for evaluating the destination component to create the received communication destination. This is not intended to limit the scope of invention in any way, but to simplify the discussion.

Figure 30A depicts a detail flowchart of operation 1300 of Figure 25 further performing the communication process within the node.

Arrow 1610 directs the flow of execution from starting operation 1300 to operation 1612. Operation 1612 performs maintaining a routing table. Arrow 5 1614 directs execution from operation 1612 to operation 1616. Operation 1616 terminates the operations of this flowchart.

Figure 30B depicts a detail flowchart of operation 1582 of Figure 29A further performing evaluating the destination component.

Arrow 1630 directs the flow of execution from starting operation 1582 to 10 operation 1632. Operation 1632 performs examining the routing table based upon the destination component to create the received communication destination. Arrow 1634 directs execution from operation 1632 to operation 1636. Operation 1636 terminates the operations of this flowchart.

The communication processor may further couple to each of the communication 15 interfaces.

Alternatively, there may be P communication processors, where P is at least two. Each communication processor couples to at least one communication interface and each communication interface couples to at least one communication processor. For each of the communication processors, and each communication 20 interface coupled to the communication, the communication processor performs the operations found in Figures 27A-27C. The subsequent operations discussed

in Figures **28A** to **30B** may be further performed by one or more of the communication processors.

The node **500** may further include a communication processor coupling mechanism coupling at least two of the communication processors.

5 Figure **31A** depicts a detail flowchart of operation **1522** of Figure **29A** further performing routing the received communication for a communication processor coupled to the communication processor coupling mechanism.

Arrow **1630** directs the flow of execution from starting operation **1522** to operation **1632**. Operation **1632** performs routing the received communication

10 based upon the communication processor coupling mechanism whenever the received communication destination is external to the node. Arrow **1634** directs execution from operation **1632** to operation **1636**. Operation **1636** terminates the operations of this flowchart.

15 Note that in certain embodiments of the invention, operation **1632** takes into account the communication processor coupling mechanism, but may not actively use that mechanism.

The node **500** may include a second communication processor coupling mechanism coupling at least two of the communication processors.

Figure **31B** depicts a detail flowchart of operation **1522** of Figure **29A** further 20 performing for each of the communication processors, routing the received communication.

Arrow **1650** directs the flow of execution from starting operation **1522** to operation **1652**. Operation **1652** performs routing the received communication based upon the communication processor coupling mechanism and based upon the second communication processor coupling mechanism whenever the received communication destination is external to the node. Arrow **1654** directs execution from operation **1652** to operation **1656**. Operation **1656** terminates the operations of this flowchart.

Figure **31C** depicts a detail flowchart of operation **1532** of Figure **28A** further performing for each of the communication processors, delivering the received communication.

Arrow **1670** directs the flow of execution from starting operation **1532** to operation **1672**. Operation **1672** performs delivering the received communication based upon the communication processor coupling mechanism whenever the received communication destination is internal to the node. Arrow **1674** directs execution from operation **1672** to operation **1676**. Operation **1676** terminates the operations of this flowchart.

Figure **32A** depicts a detail flowchart of operation **1532** of Figure **28A** further performing for each of the communication processors, the step of delivering the received communication.

Arrow **1690** directs the flow of execution from starting operation **1532** to operation **1692**. Operation **1692** performs routing-elsewhere the received communication based upon the communication processor coupling mechanism

whenever the received communication destination is internal to the node and the received communication destination is not coupled to the communication processor coupling mechanism. Arrow **1694** directs execution from operation **1692** to operation **1696**. Operation **1696** terminates the operations of this
5 flowchart.

Note that node **500** may include the communication processor coupling mechanism comprising a bus coupling at least two of the communication processors coupled by the communication processor coupling mechanism. The communication processor coupling mechanism may include a bus coupling all of
10 the communication processors coupled by the communication processor coupling mechanism. The communication processor coupling mechanism may include a bus coupling all of the communication processors.
15

In certain embodiments of the invention, failures within node **500** may in effect create two (or more) communication coupling mechanisms. The above
15 discussion of two communication coupling mechanisms may be extended to more than two communication coupling mechanisms in keeping with certain further embodiments of the invention.

A communication processor may be accessibly coupled to the routing table. The communication processor may include a finite state machine at least partially
20 controlled by a control register based upon accessing the routing table.

The communication processor may include an instruction processor accessibly coupled to a memory. A program system may reside in the accessibly coupled

memory comprised of at least the program step of processing the received communication from the communication interface, for a communication interface coupled to the communication processor as shown in operation **1432** of Figure **27B**. Note that either of the other operations of Figure **27A** and **27C** may or may not be part of the program system residing in the accessibly coupled memory of the communication processor.

The program step of **1432** of the program system residing in the accessibly coupled memory of the communication processor

Figure **32B** depicts a detail flowchart of operation **1432** of Figure **27B** further performing processing the received communication from the communication interface coupled to the communication processor.

Arrow **1710** directs the flow of execution from starting operation **1432** to operation **1712**. Operation **1712** performs determining based upon the received communication from the communication interface to create a received communication destination. Arrow **1714** directs execution from operation **1712** to operation **1716**. Operation **1716** terminates the operations of this flowchart.

Arrow **1720** directs the flow of execution from starting operation **1432** to operation **1722**. Operation **1722** performs routing the received communication whenever the received communication destination is external to the node. Arrow **1724** directs execution from operation **1722** to operation **1716**. Operation **1716** terminates the operations of this flowchart.

Arrow **1730** directs the flow of execution from starting operation **1432** to operation **1732**. Operation **1732** performs delivering the received communication whenever the received communication destination is internal to the node. Arrow **1734** directs execution from operation **1732** to operation **1716**. Operation **1716** 5 terminates the operations of this flowchart.

Figure **33A** depicts a detail flowchart of operation **1712** of Figure **32B** further performing determining based upon the received communication from the communication interface.

Arrow **1750** directs the flow of execution from starting operation **1712** to operation **1752**. Operation **1752** performs evaluating the destination component to create the received communication destination. Arrow **1754** directs execution from operation **1752** to operation **1756**. Operation **1756** terminates the operations of this flowchart.

Figure **33B** depicts a detail flowchart of operation **1612** of Figure **30A** further 15 performing maintaining the routing table.

Arrow **1770** directs the flow of execution from starting operation **1612** to operation **1772**. Operation **1772** performs generating a new routing table. Arrow **1774** directs execution from operation **1772** to operation **1776**. Operation **1776** terminates the operations of this flowchart.

20 Arrow **1780** directs the flow of execution from starting operation **1612** to operation **1782**. Operation **1782** performs distributing the new routing table to

replace the routing table. Arrow **1784** directs execution from operation **1782** to operation **1776**. Operation **1776** terminates the operations of this flowchart.

Figure **34A** depicts a detail flowchart of operation **1782** of Figure **33B** further performing distributing the new routing table.

5 Arrow **1790** directs the flow of execution from starting operation **1782** to operation **1792**. Operation **1792** performs communicating the new routing table. Arrow **1794** directs execution from operation **1792** to operation **1796**. Operation **1796** terminates the operations of this flowchart.

10 Arrow **1800** directs the flow of execution from starting operation **1782** to operation **1802**. Operation **1802** performs replacing the routing table with the new routing table. Arrow **1804** directs execution from operation **1802** to operation **1796**. Operation **1796** terminates the operations of this flowchart.

Figure **34B** depicts a detail flowchart of operation **1792** of Figure **34A** further performing communicating the new routing table.

15 Arrow **1810** directs the flow of execution from starting operation **1792** to operation **1812**. Operation **1812** performs transmitting the new routing table via each of the communication interfaces to create a new transmitted routing table. Arrow **1814** directs execution from operation **1812** to operation **1816**. Operation **1816** terminates the operations of this flowchart.

20 Arrow **1820** directs the flow of execution from starting operation **1792** to operation **1822**. Operation **1822** performs receiving the new transmitted routing

table to create the new routing table. Arrow **1824** directs execution from operation **1822** to operation **1816**. Operation **1816** terminates the operations of this flowchart.

Figure **35A** depicts a detail flowchart of operation **1802** of Figure **34A** further

5 performing replacing the routing table.

Arrow **1830** directs the flow of execution from starting operation **1802** to operation **1832**. Operation **1832** performs generating a local routing table from the new routing table. Arrow **1834** directs execution from operation **1832** to operation **1836**. Operation **1836** terminates the operations of this flowchart.

10 Arrow **1840** directs the flow of execution from starting operation **1802** to operation **1842**. Operation **1842** performs exchanging the routing table with the local routing table. Arrow **1844** directs execution from operation **1842** to operation **1836**. Operation **1836** terminates the operations of this flowchart.

Figure **35B** depicts a detail flowchart of operation **1772** of Figure **33B** further

15 performing generating the new routing table.

Arrow **1850** directs the flow of execution from starting operation **1772** to operation **1852**. Operation **1852** performs maintaining a network map. Arrow **1854** directs execution from operation **1852** to operation **1856**. Operation **1856** terminates the operations of this flowchart.

20 Arrow **1860** directs the flow of execution from starting operation **1772** to operation **1862**. Operation **1862** performs assessing the network map to create

an obstruction map. Arrow **1864** directs execution from operation **1862** to operation **1856**. Operation **1856** terminates the operations of this flowchart.

Arrow **1870** directs the flow of execution from starting operation **1772** to operation **1872**. Operation **1872** performs optimizing the network map based

5 upon the obstruction map to create the new routing table. Arrow **1874** directs execution from operation **1872** to operation **1856**. Operation **1856** terminates the operations of this flowchart.

Note that for certain nodes **500**, at least one instruction processor may be coupled to the communication processor.

10 Figure **36A** depicts a detail flowchart of operation **1532** of Figure **28A** further performing delivering the received communication.

Arrow **1890** directs the flow of execution from starting operation **1532** to operation **1892**. Operation **1892** performs delivering to the instruction processor

the received communication whenever the received communication destination targets the instruction processor. Arrow **1894** directs execution from operation

15 **1892** to operation **1896**. Operation **1896** terminates the operations of this flowchart.

A node **500** may include a communication processor coupled to a second instruction processor.

20 Figure **36B** depicts a detail flowchart of operation **1532** of Figure **28A** further performing delivering the received communication.

Arrow **1910** directs the flow of execution from starting operation **1532** to operation **1912**. Operation **1912** performs delivering to the second instruction processor the received communication whenever the received communication destination targets the second instruction processor. Arrow **1914** directs 5 execution from operation **1912** to operation **1916**. Operation **1916** terminates the operations of this flowchart.

Figure **36C** depicts a detail flowchart of operation **1382** of Figure **26** further performing processing the received communication.

Arrow **1930** directs the flow of execution from starting operation **1382** to 10 operation **1932**. Operation **1932** performs assessing the received communication from the communication interface to create a next communication activity. Arrow **1934** directs execution from operation **1932** to operation **1936**. Operation **1936** performs activating the next communication activity based upon 15 the received communication. Arrow **1938** directs execution from operation **1936** to operation **1940**. Operation **1940** terminates the operations of this flowchart.

Figure **37A** depicts a detail flowchart of operation **1932** of Figure **36C** further performing assessing the received communication.

Arrow **1950** directs the flow of execution from starting operation **1932** to operation **1952**. Operation **1952** performs extracting from the received 20 communication to create a destination component. Arrow **1954** directs execution from operation **1952** to operation **1956**. Operation **1956** terminates the operations of this flowchart.

Arrow **1960** directs the flow of execution from starting operation **1932** to operation **1962**. Operation **1962** performs evaluating the destination component to create the next communication activity. Arrow **1964** directs execution from operation **1962** to operation **1956**. Operation **1956** terminates the operations of 5 this flowchart.

Figure **37B** depicts an alternative detail flowchart of operation **1932** of Figure **36C** further performing assessing the received communication.

Arrow **1970** directs the flow of execution from starting operation **1932** to operation **1972**. Operation **1972** performs extracting from the received 10 communication to create a destination component. Arrow **1974** directs execution from operation **1972** to operation **1976**. Operation **1976** performs evaluating the destination component to create the next communication activity. Arrow **1978** directs execution from operation **1976** to operation **1980**. Operation **1980** terminates the operations of this flowchart.

15 Note that Figures **37A** and **37B** depict two of several organizational approaches to implementing very similar if not the same operations, both in accordance with certain embodiments of the invention. Figure **37A** tends to be in keeping with many event triggered real time environments, where each of the operations may be independently and concurrently performed, possibly with queuing 20 mechanisms applied before or after performance of one or more of these operations. Figure **37B** tends to be in keeping with extracting a destination component, sequentially followed by evaluating the destination component to create a next communication activity.

A node **500** may include a tunnel interface coupling to the node. The node communication process may include steps similar to the steps for interaction with a communication interface. The node **500** may include a communication processor coupled to the tunnel interface. A communication processor within the node **500** may further interact with tunnel interface in a similar manner to the interaction of the communication processor with a coupled communication interface. The interaction may be at least in part implemented as part of a program system residing in an accessibly coupled memory of an instruction processor coupled to the communication processor.

10 A communication interface within a node may include one or more ports. These ports may be used for transmitting or receiving communications within the communication pencil, or for both transmitting and receiving communications. Distinct ports within a communication interface may employ the same physical transport or distinct physical transports. The node communication process may 15 include steps similar to the steps for interaction with a communication interface. The node **500** may include a communication processor coupled to one or more ports. A communication processor within the node **500** may further interact with ports in a similar manner to the interaction of the communication processor with a coupled communication interface. The interaction may be implemented as part 20 of a program system residing in an accessibly coupled memory of an instruction processor coupled to the communication processor.

Figure **38** depicts a communication interface **900** including P1=4 input ports and P2=4 output ports coupled to a communication pencil including optical fibers **902**

and **904**, each optical fiber handling one way traffic with optical fiber **902** coupled **940** through optronic amplifier **942** coupling **944** to optical fiber **904**, in accordance with certain embodiments of the invention.

As used herein optronic refers to devices, technologies and communications 5 protocols which may include some or all of the visible light or near visible light spectrum. Infra-red and ultraviolet spectral ranges will be considered as part of the near visible light spectrum.

As used herein, optronic amplifier **942** may include a splitter feeding multiple optronic signals, each optronic signal being band pass filtered to a color 10 component signal, each color component signal being separately amplified, then combined to create the optronic signal injected **944** to optical cable **904**. The separate amplification of each color component signal may further include separate gain controls providing different color component signal gain across the respective color component signal amplifiers. The separate amplification of each 15 color component signal may also include splitting the color component signal to create multiple split color signals, filtering each of the split color signals, separate amplification of the filtered, split color signals, which are then combined to form the amplified color component signal.

The P2=4 output ports **930-936** control a circuit which couples **906** to output 20 optical fiber **902**. Note that P2 may be 1. P2 may be greater than 1. P2 may be greater than 4. The discussion is presented with P2=4 to both demonstrate the scalability of the communication interface in terms of output ports and to constrain the complexity of the discussion.

The P1=4 input ports **958, 968, 978** and **988** are driven from a circuit coupled **946** to input optical fiber **904**. Note that P1 may be 1. P1 may be greater than 1. P1 may be greater than 4. The discussion is presented with P1=4 to both demonstrate the scalability of the communication interface in terms of input ports 5 and to constrain the complexity of the discussion.

Output port **930** controls optronic source **910** generating optronic signal **912**.

Output port **932** controls optronic source **914** generating optronic signal **916**.

Output port **934** controls optronic source **918** generating optronic signal **920**.

Output port **936** controls optronic source **922** generating optronic signal **924**.

10 Optronic signals **912, 916, 920** and **924** are presented to combiner **908** to create a combined optronic signal coupled **906** to output optical fiber **902**. Coupling **906** to output optical fiber **902** may include injection of the combined optronic signal at a bend in optical fiber **902**. Coupling **906** to output optical fiber **902** may further include injection of the combined optronic signal at a bend in optical fiber 15 **902** through an optronic amplifier. The optronic amplifier may include gain controls.

Input optical fiber **904** couples **946** to splitter **948** to generate optronic signals **950, 960, 970**, and **980**. Optronic signal **950** feeds filter **952** to create optronic color signal **954** which stimulates optronic detector **956**. Optronic detector **956** 20 drives the state of port **958**. Optronic signal **960** feeds filter **962** to create optronic color signal **964** which stimulates optronic detector **966**. Optronic detector **966** drives the state of port **968**. Optronic signal **970** feeds filter **972** to create optronic color signal **974** which stimulates optronic detector **976**. Optronic detector **976**

drives the state of port **978**. Optronic signal **980** feeds filter **982** to create optronic color signal **984** which stimulates optronic detector **986**. Optronic detector **986** drives the state of port **988**.

Coupling **946** between output optical fiber **904** and splitter **948** may include a
5 receptor positioned at a bend in optical fiber **904**. Coupling **946** from output optical fiber **904** may further include the receptor containing an optronic amplifier positioned at the bend in optical fiber **904** to generate the optronic signal presented to splitter **948**. The optronic amplifier may include gain controls.

The preceding embodiments have been provided by way of example and are not

10 meant to constrain the scope of the following claims.

Claims

1. A communications network with M orthogonal directions supporting communications between an M dimensional lattice of a multiplicity of nodes each 5 containing a multiplicity of ports, said communications network comprising:

a communication grid interconnecting said nodes, said grid further comprising:

a multiplicity of communication pencils, for each of said M orthogonal directions;

10 wherein each of said communication pencils in each orthogonal direction is coupled with a corresponding node pencil containing a multiplicity of nodes to couple each of said nodes of said corresponding node pencil directly to the other nodes of said corresponding node pencil;

wherein M is at least two; and

15 wherein the number of nodes in each of said node pencils in a first of said orthogonal directions is at least four;

wherein the number of nodes in each of said node pencils in a second of said orthogonal directions is at least two.

2. The communications network of Claim 1,

20 wherein the number of nodes in each of said node pencils in said second orthogonal direction is at least three.

3. The communications network of Claim 2,

wherein the number of nodes in each of said node pencils in said second orthogonal direction is at least four.

4. The communications network of Claim 1,

wherein each of said communication pencils is comprised of the number 5 of communications paths required to interconnect each node of said corresponding node pencil directly to the other of said nodes of said corresponding node pencil.

5. The communications network of Claim 4,

wherein each of said nodes is comprised of P coupled communications 10 processors;

wherein P is at least two; and

wherein each of said communications processors is coupled to at least one of said communication pencils.

6. The communications network of Claim 5,

15 wherein P is a factor of $M^*(N-1)$.

7. The communications network of Claim 6,

wherein each of said coupled communications processors comprises at least $M^*(N-1)/P$ ports.

8. The communications network of Claim 5,

20 wherein at least one of said communications processors is further comprised of at least one instruction processor accessibly coupled to a memory.

9. The communications network of Claim 5,

wherein each of said communications processors is further comprised of at least one instruction processor accessibly coupled to a memory.

10. The communications network of Claim 8,

5 wherein each of said communications processors is comprised of a communications instruction processor accessibly coupled to said memory; and

wherein said communications instruction processor is communicatively coupled to at least one of said ports.

11. The communications network of Claim 10,

10 wherein said instruction processor acts as said communications instruction processor.

12. The communications network of Claim 10,

wherein each of said communications instruction processor and said instruction processor reside in a single package.

15 13. The communications network of Claim 10,

wherein said accessibly coupled memory comprises a memory module.

14. The communications network of Claim 10,

wherein each of said nodes further comprises a package containing at least two of said communications processors.

20 15. The communications network of Claim 14,

wherein at least one of said communications processors in said package comprises an instruction processor accessibly coupled to at least one memory.

16. The communications network of Claim 15,

wherein said package comprises said accessibly coupled memory.

5 17. The communications network of Claim 15,

wherein said accessibly coupled memory is comprised of an external memory circuit accessibly coupled to at least one of said instruction processors.

18. The communications network of Claim 8,

wherein said communications processors are coupled by a bus.

10 19. The communications network of Claim 18,

wherein at least one of said nodes is further comprised of a bus arbitration scheme controlling said bus coupling.

20. The communications network of Claim 19,

wherein said bus arbitration scheme controlling said bus coupling supports
15 a bus master.

21. The communications network of Claim 20,

wherein said bus master is one of said communications processors.

22. The communications network of Claim 21,

wherein said bus master can over time be any of said communications
20 processors.

23. The communications network of Claim 6,

wherein $N-1$ is a factor of P .

24. The communications network of Claim 23,
wherein P is equal to $N-1$.

25. The communications network of Claim 6,
5 wherein M is a factor of P .

26. The communications network of Claim 6,
wherein P is equal to M .

27. The communications network of Claim 5,
wherein said communications processors are coupled by a direct
10 connection network to each of said communications processors coupled directly
to each of the remaining of said communications processors.

28. The communications network of Claim 4,
wherein each of said nodes comprises $M^*(N-1)$ ports.

29. The communications network of Claim 28,
15 wherein at least one of said nodes comprises more than $M^*(N-1)$ ports.

30. The communications network of Claim 4,
wherein at least one of said nodes is comprised of a coupled
communications processor;
wherein said communications processor is coupled to each of said
20 communication pencils.

31. A method of communicating between a first node and a second node wherein said first node is coupled to a first communication pencil coupled to a third node and said third node is coupled to a second communication pencil coupled to said second node, wherein each of said communication pencils includes at least one physical transport layer, comprising the steps of:

5 said first node communicating with said third node via said first communication pencil further comprised of the step of traversing all necessary physical transport layers of said physical transport layers included in said first communication pencil; and

10 said third node communicating with said second node via said second communication pencil further comprised of the step of traversing all necessary physical transport layers of said physical transport layers included in said second communication pencil.

32. A method of Claim 31,

15 wherein each of said coupled communication pencils includes one of said physical transport layers.

33. A method of Claim 31,

 wherein at least one of said communication pencils includes a first and a second of said physical transport layers; and

20 wherein the step of traversing said physical transport layers for said communication pencil including said first physical transport layer and said second physical transport layer comprises of the steps of:

 traversing said first physical transport layer;

traversing between said first physical transport layer and said second physical transport layer; and

traversing said second physical transport layer.

34. A method of Claim 31,

5 wherein said physical transport layer includes a wave guide.

35. A method of Claim 34,

wherein said wave guide physically transports at least in said microwave domain.

36. A method of Claim 34,

10 wherein said wave guide physically transports at least in said infrared domain.

37. A method of Claim 34,

wherein said wave guide physically transports at least in said optical domain.

15 38. A method of Claim 37,

wherein said wave guide includes at least one optical fiber.

39. A method of Claim 34,

wherein said wave guide physically transports at least in said radio domain.

20 40. A method of Claim 31,

wherein said physical transport layer includes at least one coaxial cable.

41. A method of Claim 31,

wherein said physical transport layer includes at least one wire.

42. A method of Claim 41,

wherein said physical transport layer includes at least one twisted pair of

5 wires.

43. A node coupling to M communication pencils, wherein M is at least two,

comprising:

M communication interfaces, each of said communication interfaces coupling to a corresponding communication pencil;

10 wherein a communication process performed within said node controlling all of said communications interfaces is comprised of:

 interacting within said node with said communication interface, for each of said communication interfaces, further comprising the steps of:

 receiving a first communication from said communication interface

15 to create a received communication from said communication interface;

 processing said received communication from said communication interface; and

 sending a local communication to said communication interface to create a second communication to said communication interface.

20 44. The node of Claim 43, further comprising:

 a communication processor coupled to at least one of said communication interfaces.

45. The node of Claim 44,

wherein said communication processor couples to each of said communication interfaces.

46. The node of Claim 44,

wherein, for each of said communication interfaces coupled to said communication processor, the step of receiving said first communication is further comprised of said communication processor receiving said first communication from said communication interface to create said received communication from said communication interface;

10 wherein, for each of said communication interfaces coupled to said communication processor, the step of processing said received communication is further comprised of said communication processor processing said received communication from said communication interface; and

15 wherein, for each of said communication interfaces coupled to said communication processor, the step of sending said local communication is further comprised of said communication processor sending said local communication to said communication interface to create said second communication to said communication interface.

47. The node of Claim 46,

20 wherein the step of processing said received communication is further comprised of the steps of:

determining a received communication destination based upon said received communication from said communication interface;

routing said received communication whenever said received communication destination is external to said node; and

delivering said received communication whenever said received communication destination is internal to said node.

5 48. The node of Claim 47,

wherein the step of determining said received communication destination is further comprised of the steps of:

extracting from said received communication to create a destination component; and

10 evaluating said destination component to create said received communication destination.

49. The node of Claim 48,

wherein said communication process within said node is further comprised the step of:

15 maintaining a routing table;

wherein the step of evaluating said destination component is further comprised of the step of:

examining said routing table based upon said destination component to create said received communication destination.

20 50. The node of Claim 49, further comprising:

P of said communication processors, each of said communication processors couples to at least one of said communication interfaces;

wherein P is at least two;

wherein for each of said communication processors, for each of said communication interfaces coupled to said communications processor, said communication processor performs steps comprising the steps of:

receiving said first communication from said communication interface to
5 create a received communication from said communication interface;

processing said received communication from said communication interface; and

sending said local communication to said communication interface to
create said second communication to said communication interface.

10 51. The node of Claim 50, further comprising:

a communication processor coupling mechanism coupling at least two of
said communication processors.

52. The node of Claim 51,

wherein for each of said communication processors coupled by said
15 communication processor coupling mechanism, the step of routing said received
communication is further comprised of the step of:

routing said received communication based upon said communication
processor coupling mechanism whenever said received communication
destination is external to said node.

20 53. The node of Claim 51, further comprising:

a second communication processor coupling mechanism coupling at least
two of said communication processors;

wherein for each of said communication processors, the step of routing said received communication is further comprised of the step of:

routing said received communication based upon said communication processor coupling mechanism and based upon said second communication processor coupling mechanism whenever said received communication destination is external to said node.

54. The node of Claim 51,

wherein for each of said communication processors, the step of delivering said received communication is further comprised of the step of:

10 delivering said received communication based upon said communication processor coupling mechanism whenever said received communication destination is internal to said node.

55. The node of Claim 51,

wherein for each of said communication processors not coupled by said communication processor coupling mechanism, the step of delivering said received communication is further comprised of the steps of:

routing-elsewhere said received communication based upon said communication processor coupling mechanism whenever said received communication destination is internal to said node and said received communication destination is not coupled to said communication processor coupling mechanism.

20 56. The node of Claim 51,

wherein said communication processor coupling mechanism couples all of said communication processors.

57. The node of Claim 51,

wherein said communication processor coupling mechanism is further 5 comprised of a bus coupling at least two of said communication processors coupled by said communication processor coupling mechanism.

58. The node of Claim 57,

wherein said communication processor coupling mechanism is further comprised of a bus coupling all of said communication processors coupled by 10 said communication processor coupling mechanism.

59. The node of Claim 58,

wherein said communication processor coupling mechanism is further comprised of a bus coupling all of said communication processors.

60. The node of Claim 51,

wherein said communication processor coupling mechanism is further 15 comprised of a direct coupling of each pair of said communication processors coupled by said communication processor coupling mechanism.

61. The node of Claim 60,

wherein said communication processor coupling mechanism couples all of 20 said communication processors.

62. The node of Claim 49,

wherein said communication processor is accessibly coupled to said routing table.

63. The node of Claim 62,

wherein said communication processor is further comprised of a finite state engine accessibly coupled to said routing table.

64. The node of Claim 63,

wherein said finite state engine is at least partially controlled by a control register based upon accessing said routing table.

65. The node of Claim 49,

wherein said communication processor is further comprised of an instruction processor accessibly coupled to a memory;

wherein said accessibly coupled memory contains a program system comprised of said program step of:

processing said received communication from said communication interface further comprised of said program steps of:

determining based upon said received communication from said communication interface further comprised of said program step of: evaluating said destination component to create said received communication destination;

routing said received communication whenever said received communication destination is external to said node; and

delivering said received communication whenever said received communication destination is internal to said node.

66. The node of Claim 49,

wherein the step of maintaining said routing table is further comprised of at least one of said collection comprising the steps of:

generating a new routing table; and

5 distributing said new routing table to replace said routing table.

67. The node of Claim 66,

wherein the step of distributing said new routing table is further comprised of the steps of:

communicating said new routing table; and

10 replacing said routing table with said new routing table.

68. The node of Claim 66,

wherein the step of communicating said new routing table is further comprised at least one of the collection comprised of the steps of:

transmitting said new routing table via each of said communication

15 interfaces to create a new transmitted routing table; and

receiving said new transmitted routing table to create said new routing table.

69. The node of Claim 67,

wherein the step of replacing said routing table is further comprised of the

20 steps of:

generating a local routing table from said new routing table; and

exchanging said routing table with said local routing table.

70. The node of Claim 66,

wherein the step of generating said new routing table is further comprised of the steps of:

5 maintaining a network map;

assessing said network map to create an obstruction map; and

optimizing said network map based upon said obstruction map to create
said new routing table.

71. The node of Claim 70,

wherein said network map includes said obstruction map.

10 72. The node of Claim 47, further comprising:

at least one instruction processor coupled to said communication
processor; and

wherein the step of delivering said received communication is further
comprised of the steps of:

15 delivering to said instruction processor said received communication
whenever said received communication destination targets said instruction
processor.

73. The node of Claim 72,

a second instruction processor coupled to said communication processor;

20 and

wherein the step of delivering said received communication is further
comprised of the steps of:

delivering to said second instruction processor said received communication whenever said received communication destination targets said second instruction processor.

74. The node of Claim 46,

5 wherein the step of processing said received communication is further comprised of the steps of:

assessing said received communication from said communication interface to create a next communication activity;

activating said next communication activity based upon said received 10 communication.

75. The node of Claim 74,

wherein the step of assessing said received communication is further comprised of the steps of:

extracting from said received communication to create a destination 15 component; and

evaluating said destination component to create said next communication activity.

76. The node of Claim 75,

wherein said communication process within said node is further comprised

20 the step of:

maintaining a routing table;

wherein the step of evaluating said destination component to create said received communication destination is further comprised of the steps of:

examining said routing table based upon said destination component to create said next communication activity.

77. The node of Claim 74,

wherein the step of activating said next communication activity is further

5 comprised of at least one of said members of said collection comprising the steps of:

routing said received communication external to said node; and

delivering said received communication internal to said node.

78. The node of Claim 46, further comprising:

10 a tunnel interface coupling said communication processor to a communications tunnel; and

wherein said communication processor, performs steps comprising the steps of:

receiving a first communication from said tunnel interface to create a

15 received communication from said tunnel interface;

processing said received communication from said tunnel interface; and

sending a local communication to said tunnel interface to create a second communication.

79. The node of Claim 46,

20 wherein at least one of said communication interfaces is further comprised of at least two ports coupled to said coupled communication pencil of said communication interface.

80. The node of Claim 79,

wherein for each communication interface further comprised of ports, for at least one port coupled to said coupled communication pencil, the step of receiving said first communication from said communication interface is further comprised of the step of:

receiving said first communication from said port to create a received communication from said port of said communication interface.

81. The node of Claim 79,

wherein for each communication interface further comprised of ports, for at least one port coupled to said coupled communication pencil, the step of sending said local communication to said communication interface is further comprised of the step of:

sending said local communication to said port to create said second communication.

15 82. The node of Claim 46,

wherein a first of said communication pencils includes a bus; and wherein said communication interface coupling to said first communication pencil includes a bus interface to said bus.

83. The node of Claim 44, further comprising

20 wherein, for at least one of said communication interfaces, said node performing the step of receiving said first communication is further comprised of said communication processor performing the step of receiving said first

communication from said communication interface to create said received communication from said communication interface.

84. The node of Claim 44, further comprising

wherein, for at least one of said communication interfaces, said node performing the step of processing said received communication is further comprised of said communication processor performing the step of processing said received communication from said communication interface.

85. The node of Claim 44, further comprising

wherein, for at least one of said communication interfaces, said node performing the step of sending said local communication is further comprised of said communication processor performing the step of sending said local communication to said communication interface to create said second communication.

86. The node of Claim 43, further comprising:

a tunnel interface coupling said node to a communications tunnel; and wherein said node, performs steps comprising the steps of:
receiving a first communication from said tunnel interface to create a received communication from said tunnel interface;
processing said received communication from said tunnel interface; and
20 sending a local communication to said tunnel interface to create a second communication.

87. The node of Claim 43,

wherein at least one of said communication interfaces is further comprised of at least two ports coupled to said coupled communication pencil of said communication interface.

88. The node of Claim 87,

5 wherein for each communication interface further comprised of ports, for at least one port coupled to said coupled communication pencil, the step of receiving said first communication from said communication interface is further comprised of the step of:

10 receiving said first communication from said port to create a received communication from said port of said communication interface.

89. The node of Claim 87,

wherein for each communication interface further comprised of ports, for at least one port coupled to said coupled communication pencil, the step of sending said local communication to said communication interface is further comprised of the step of:

15 sending said local communication to said port to create said second communication.

90. The node of Claim 43,

20 wherein the step of processing said received communication is further comprised of the steps of:

determining based upon said received communication from said communication interface to create a received communication destination;

routing said received communication whenever said received communication destination is external to said node; and

delivering said received communication whenever said received communication destination is internal to said node.

5 91. The node of Claim 90,

wherein the step of determining based upon said received communication to create said received communication destination is further comprised of the steps of:

10 extracting from said received communication to create a destination component; and

evaluating said destination component to create said received communication destination.

92. The node of Claim 91,

wherein said communication process within said node is further comprised 15 the step of:

maintaining a routing table;

wherein the step of evaluating said destination component to create said received communication destination is further comprised of the steps of:

examining said routing table based upon said destination component to 20 create said received communication destination.

SYSTEM, METHOD, AND NODE OF THE MULTI-DIMENSIONAL PLEX COMMUNICATION NETWORK

Abstract

A method of communicating is disclosed and a communications network having

5 M orthogonal directions that supports communications between an M dimensional lattice of nodes, where M is at least two with greater communication performance than mesh or toroidal communications schemes and lower system complexity than either direct interconnect or switch interconnect schemes.

AGLE0003

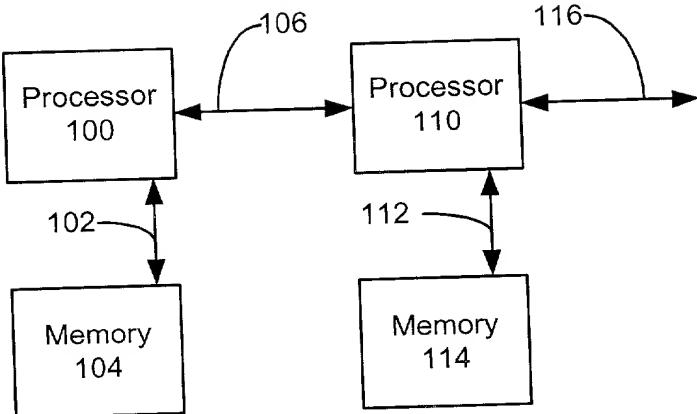


Fig. 1A
Prior Art

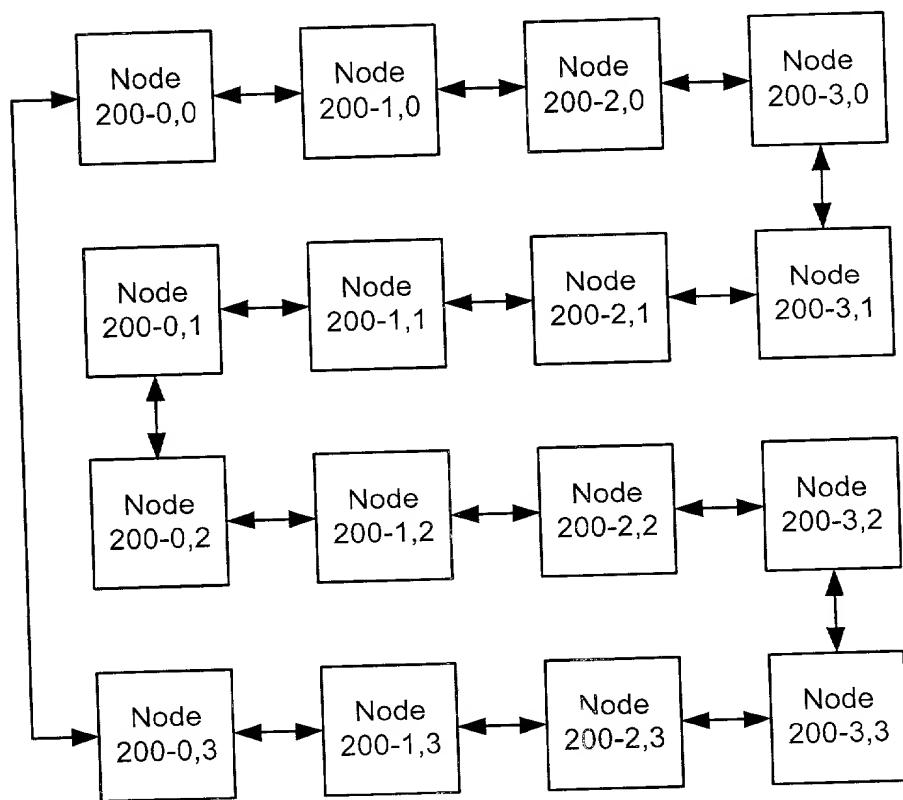


Fig. 1B
Prior Art

AGLE0003

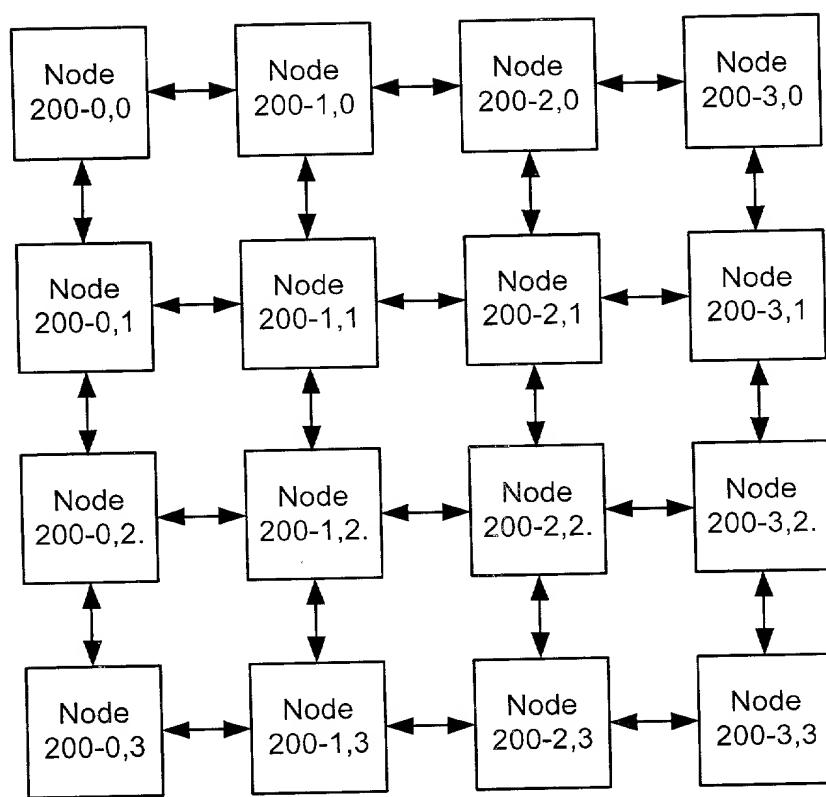


Fig. 1C
Prior Art

AGLE0003

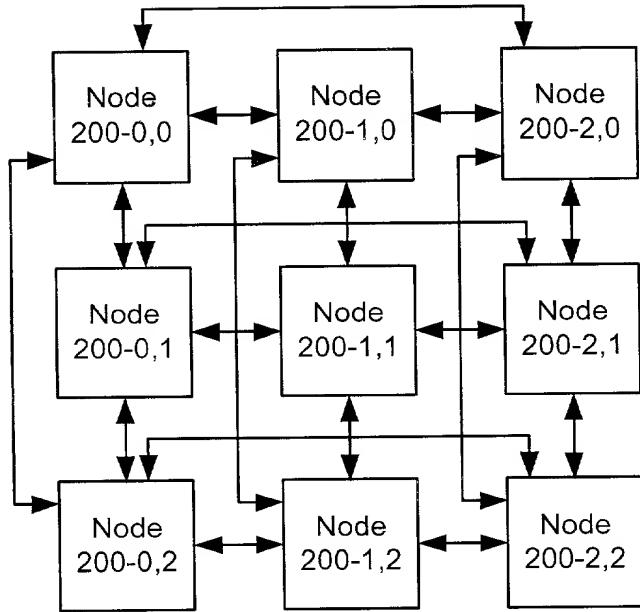


Fig. 2A Prior Art

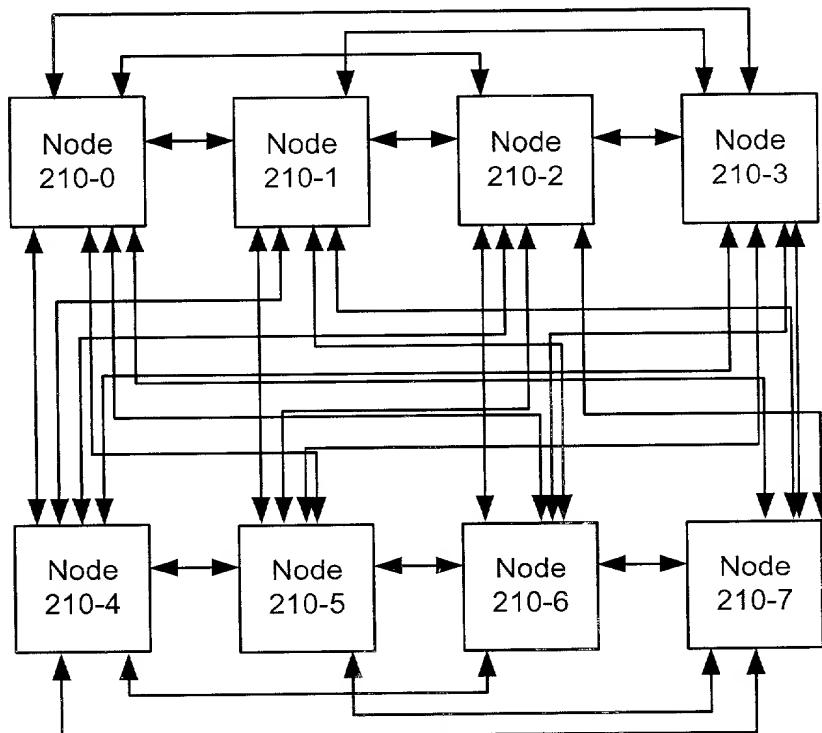
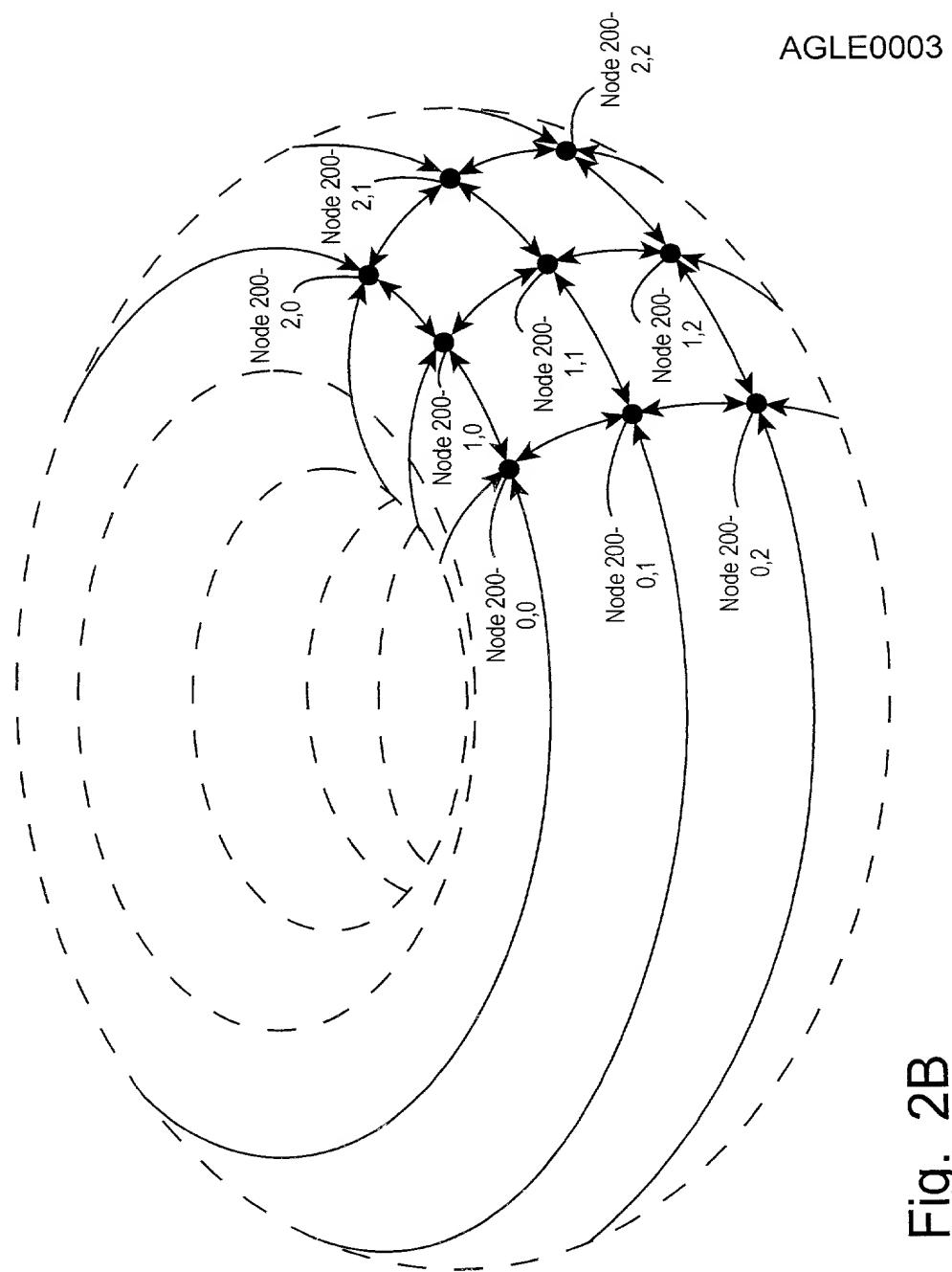


Fig. 2C Prior Art

Fig. 2B



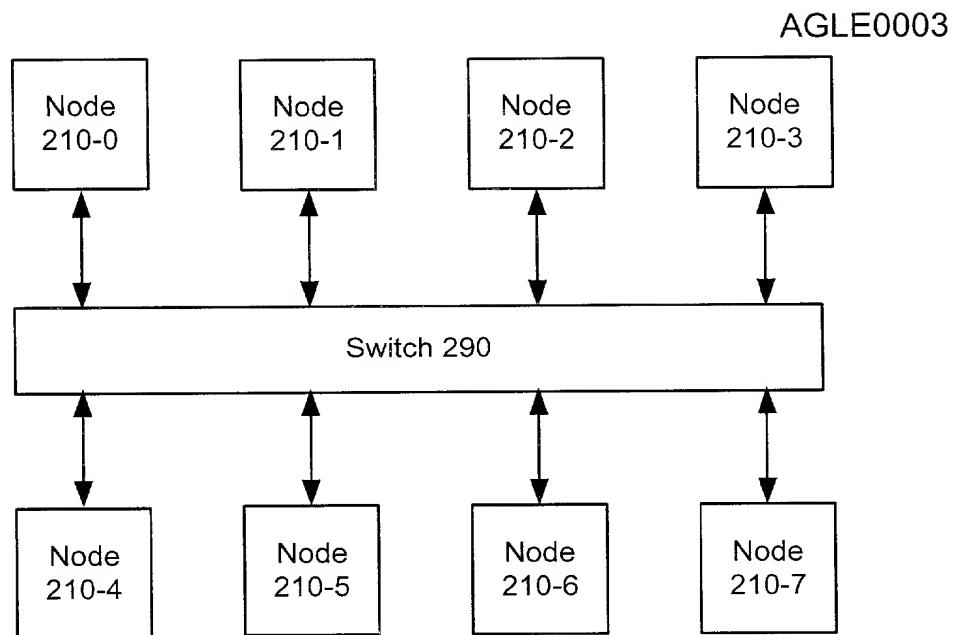


Fig. 3A Prior Art

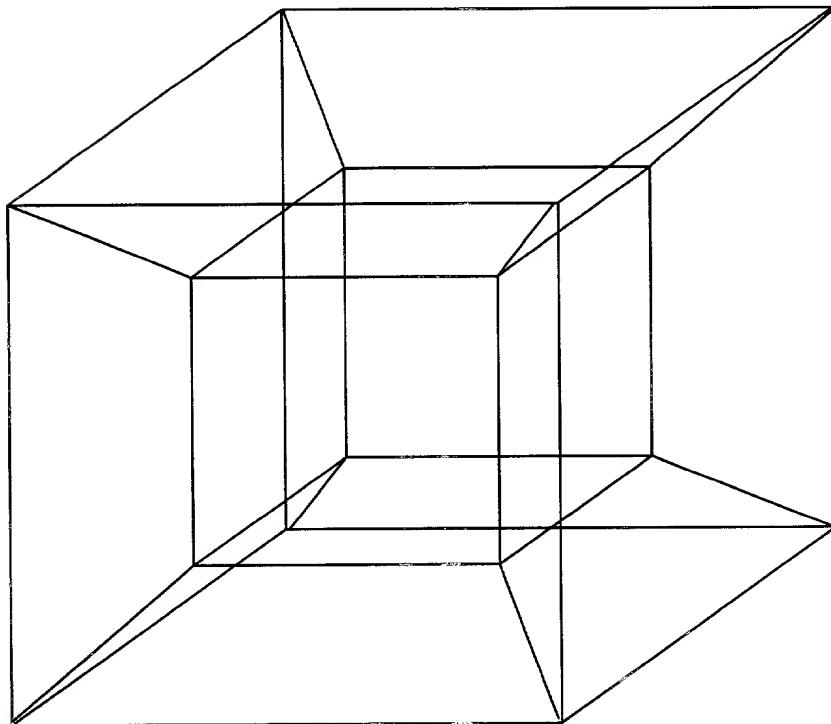


Fig. 3B
Prior Art

AGLE0003

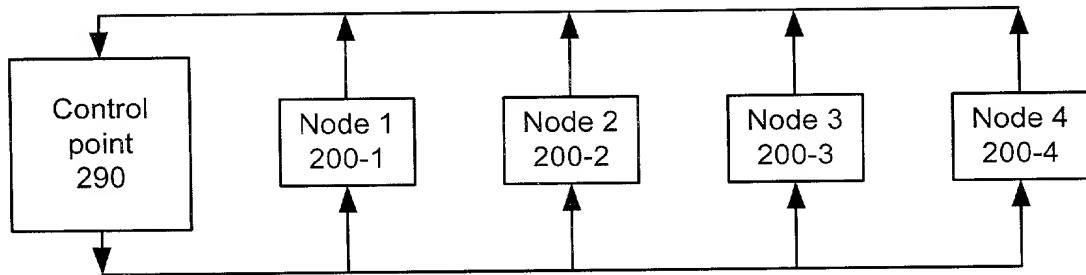


Fig. 3C
Prior Art

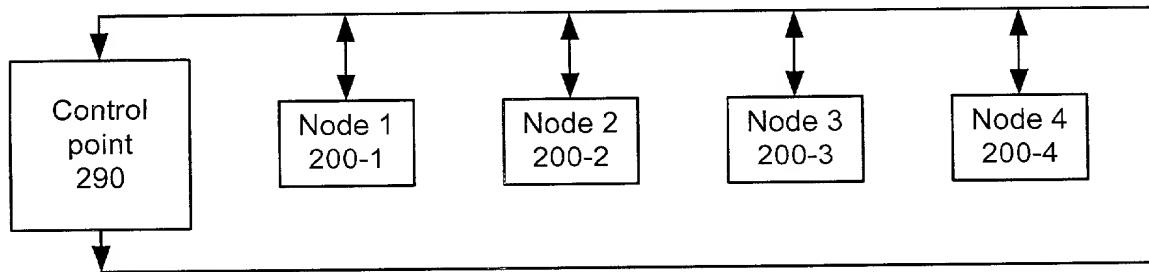


Fig. 3D
Prior Art

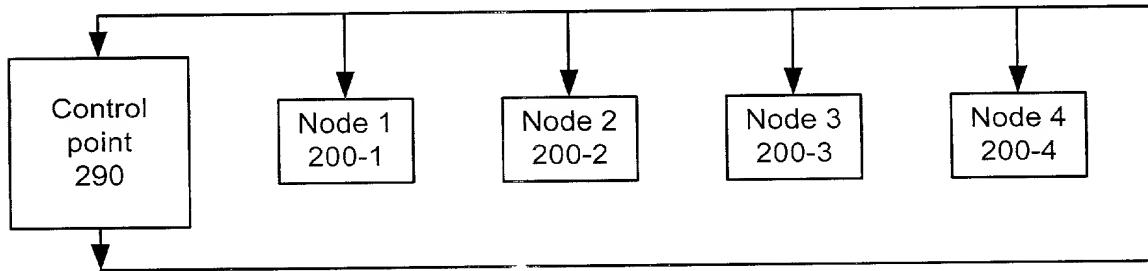


Fig. 3E
Prior Art

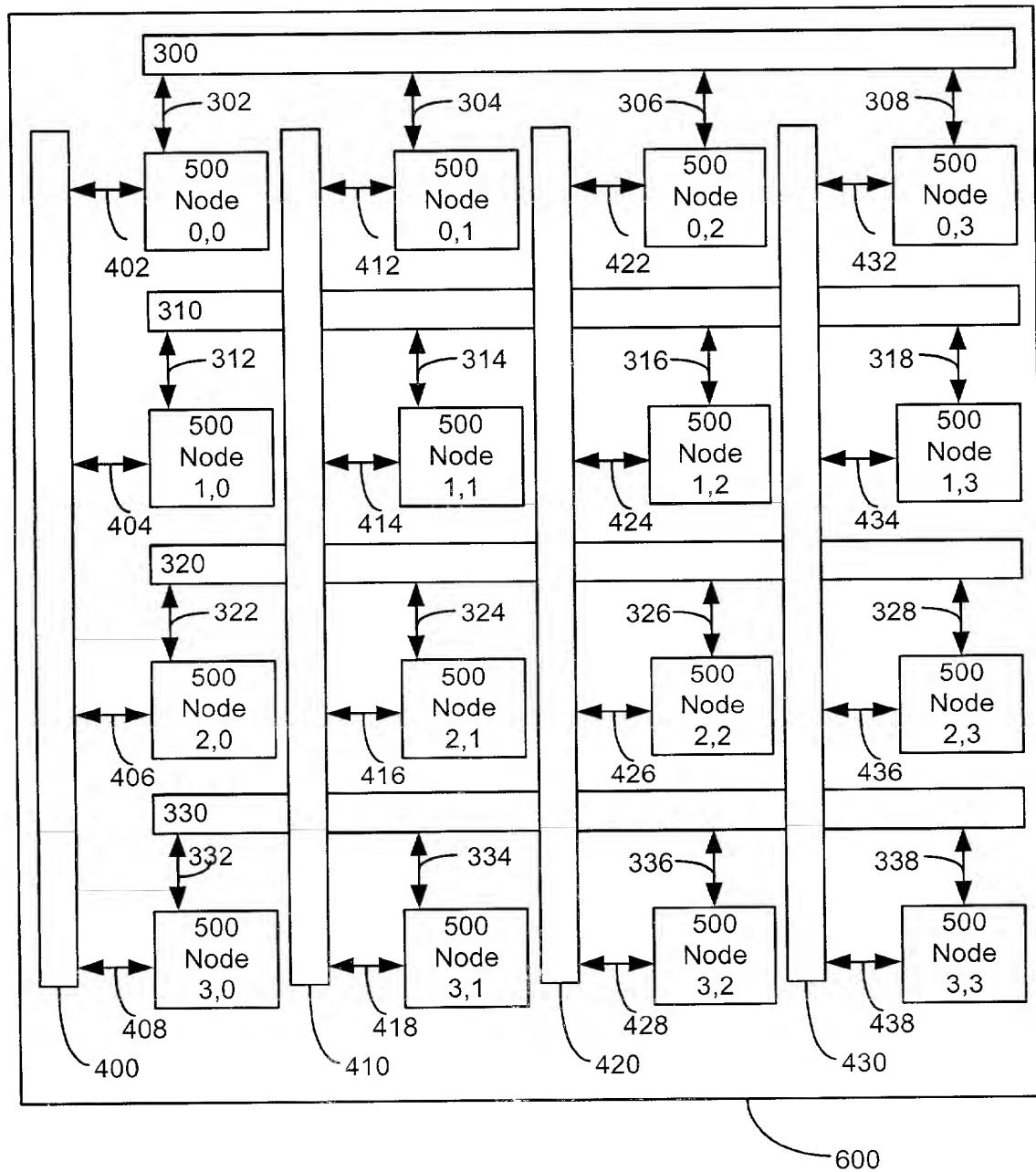


Fig. 4A

AGLE0003

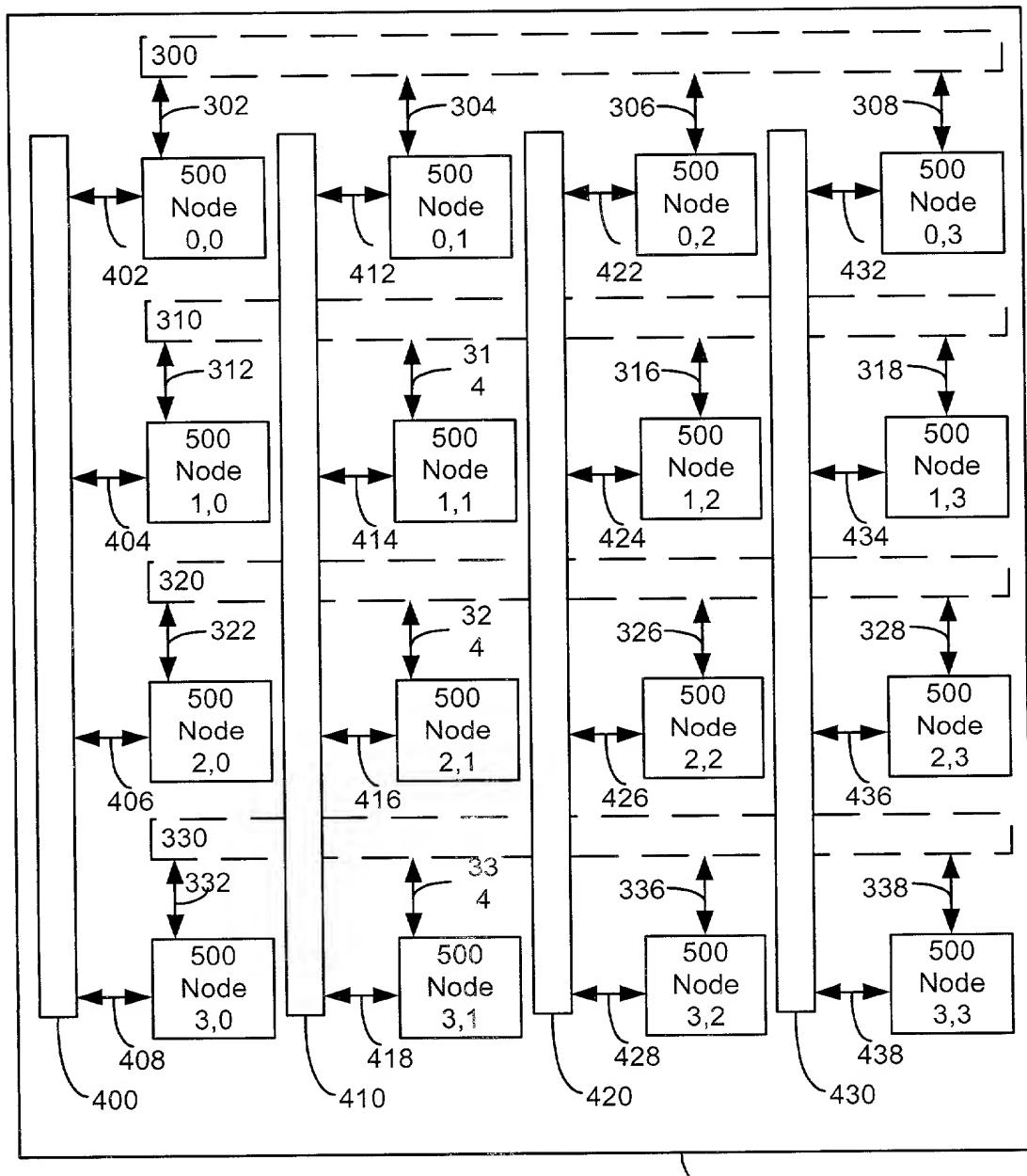


Fig. 4B

AGLE0003

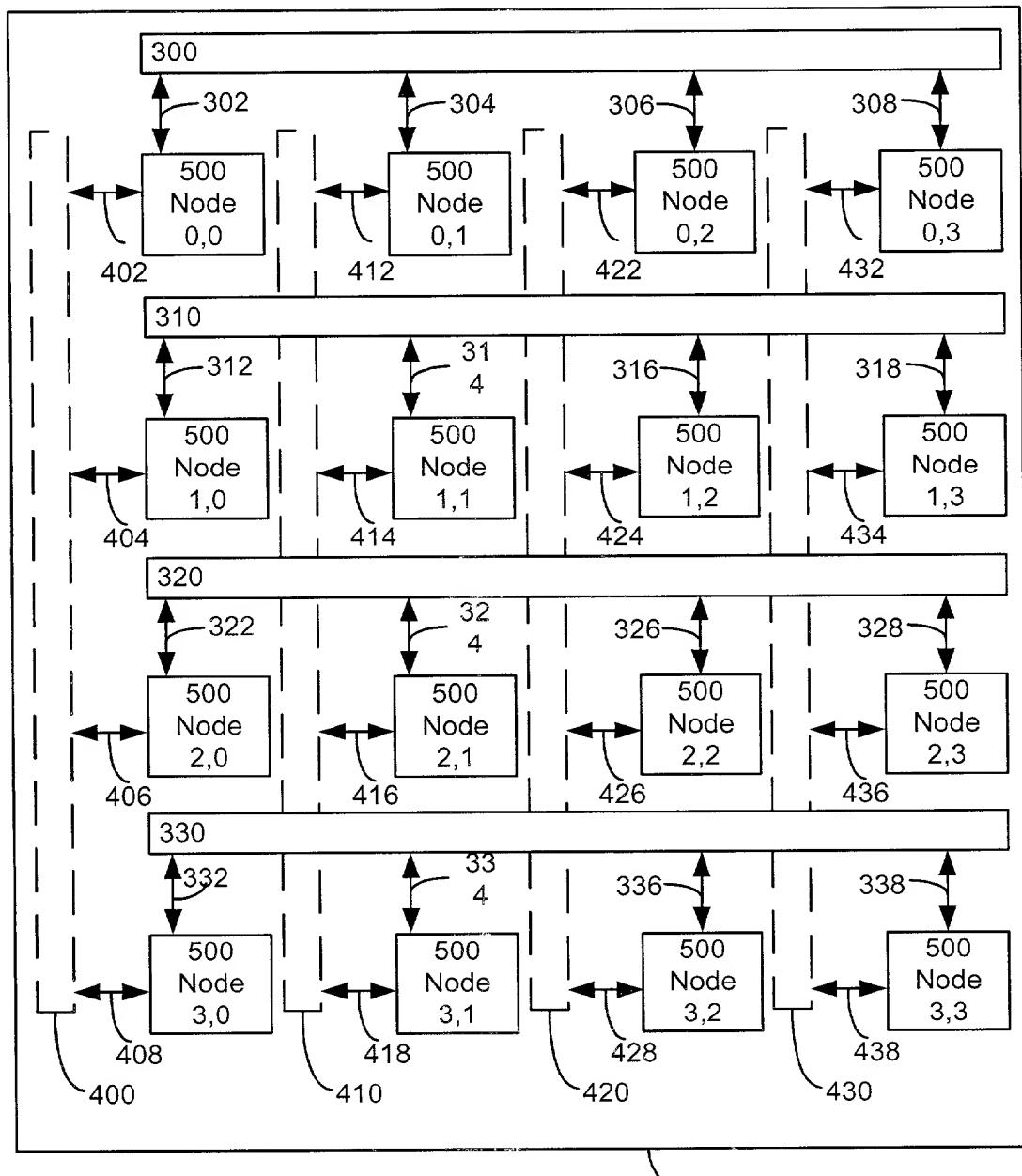


Fig. 4C

AGLE0003

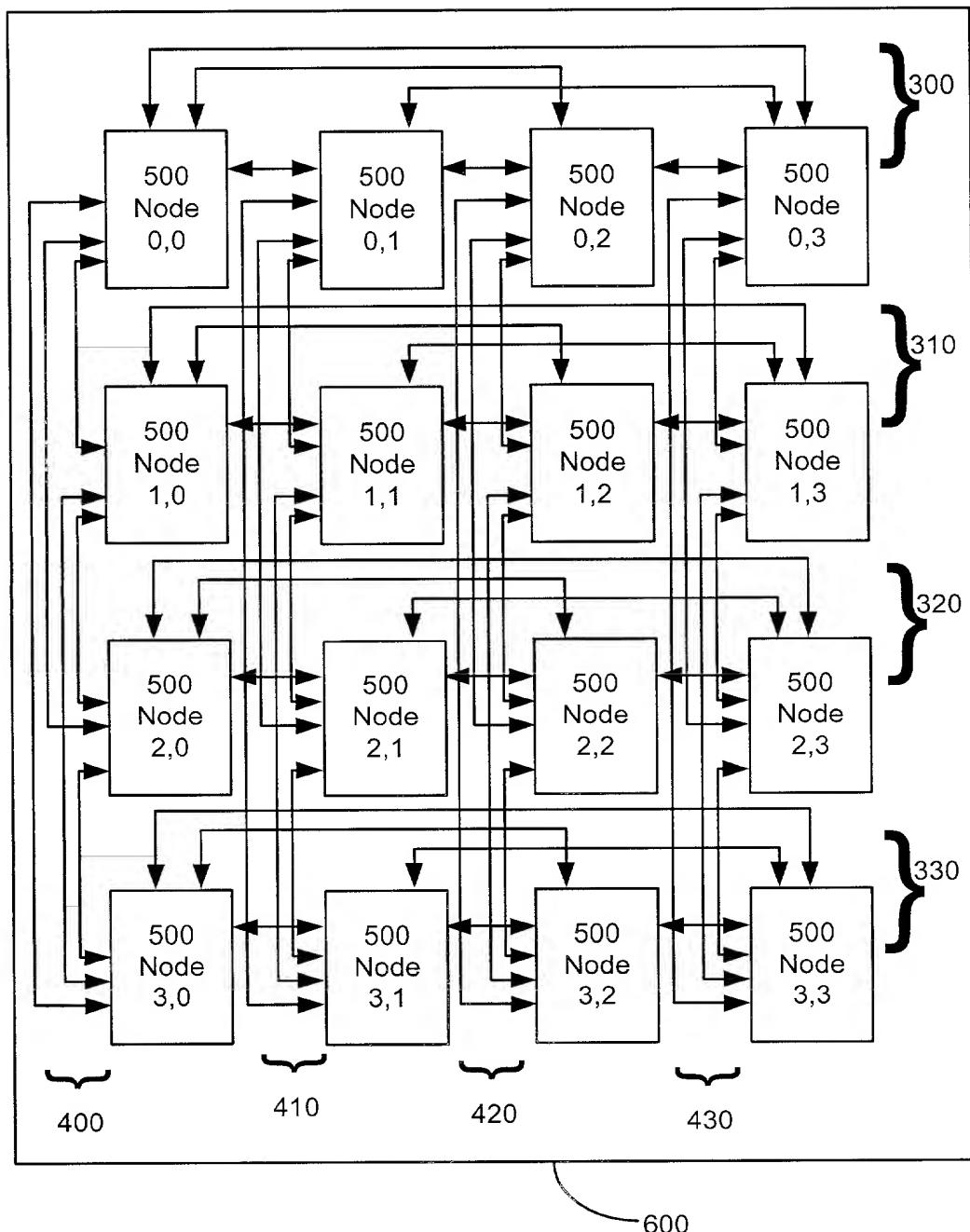


Fig. 4D

AGLE0003

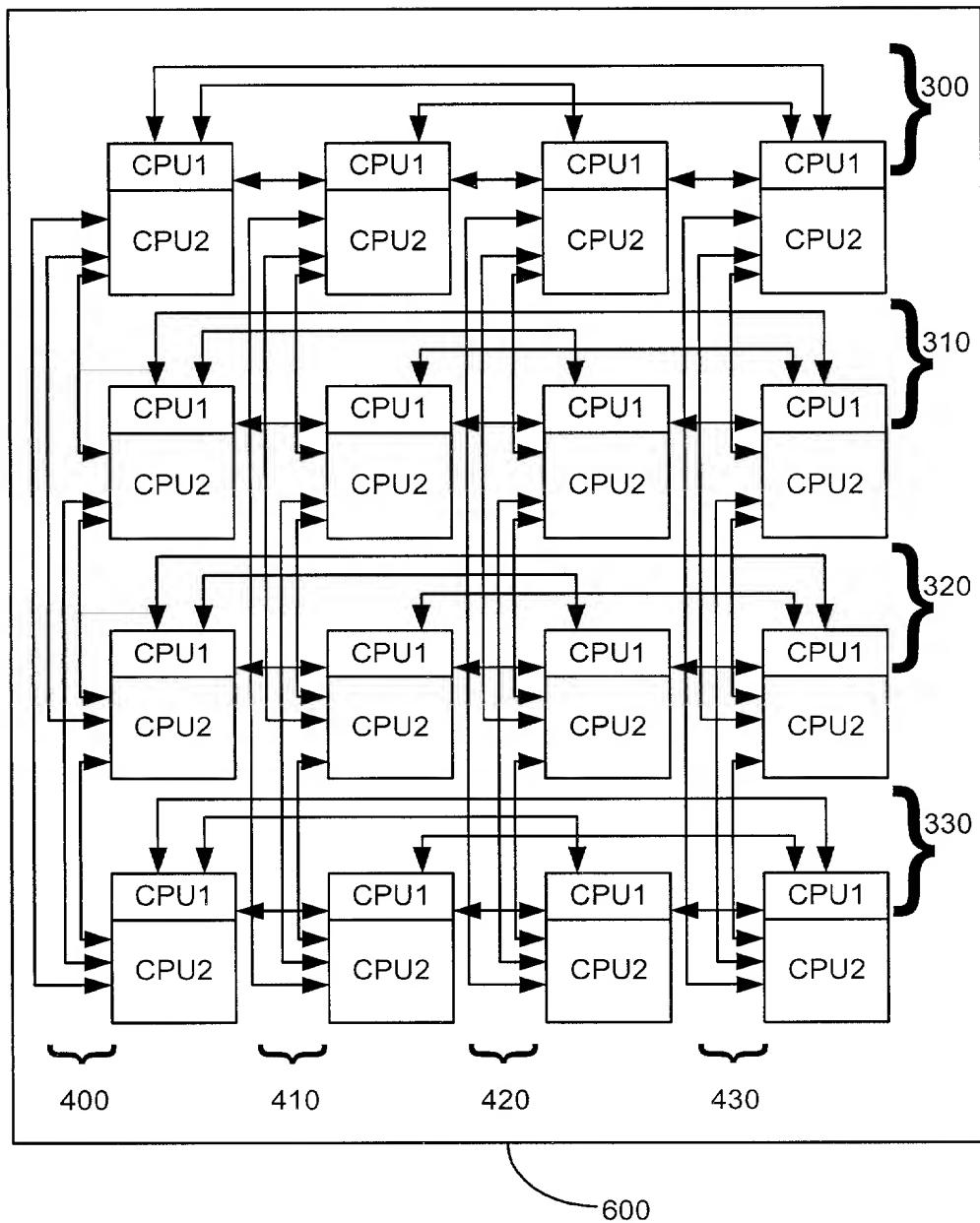


Fig. 5

AGLE0003

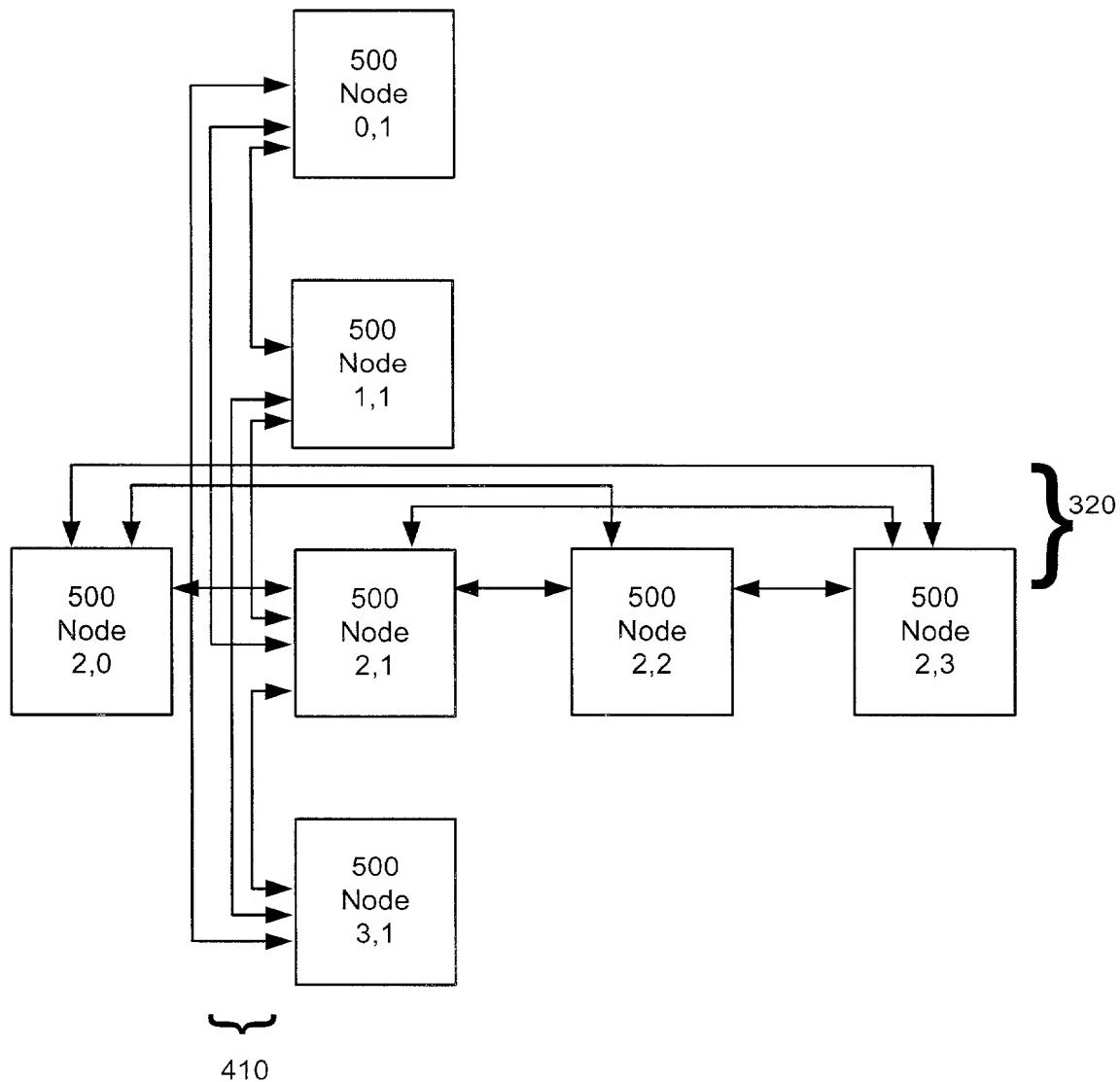


Fig. 6

AGLE0003

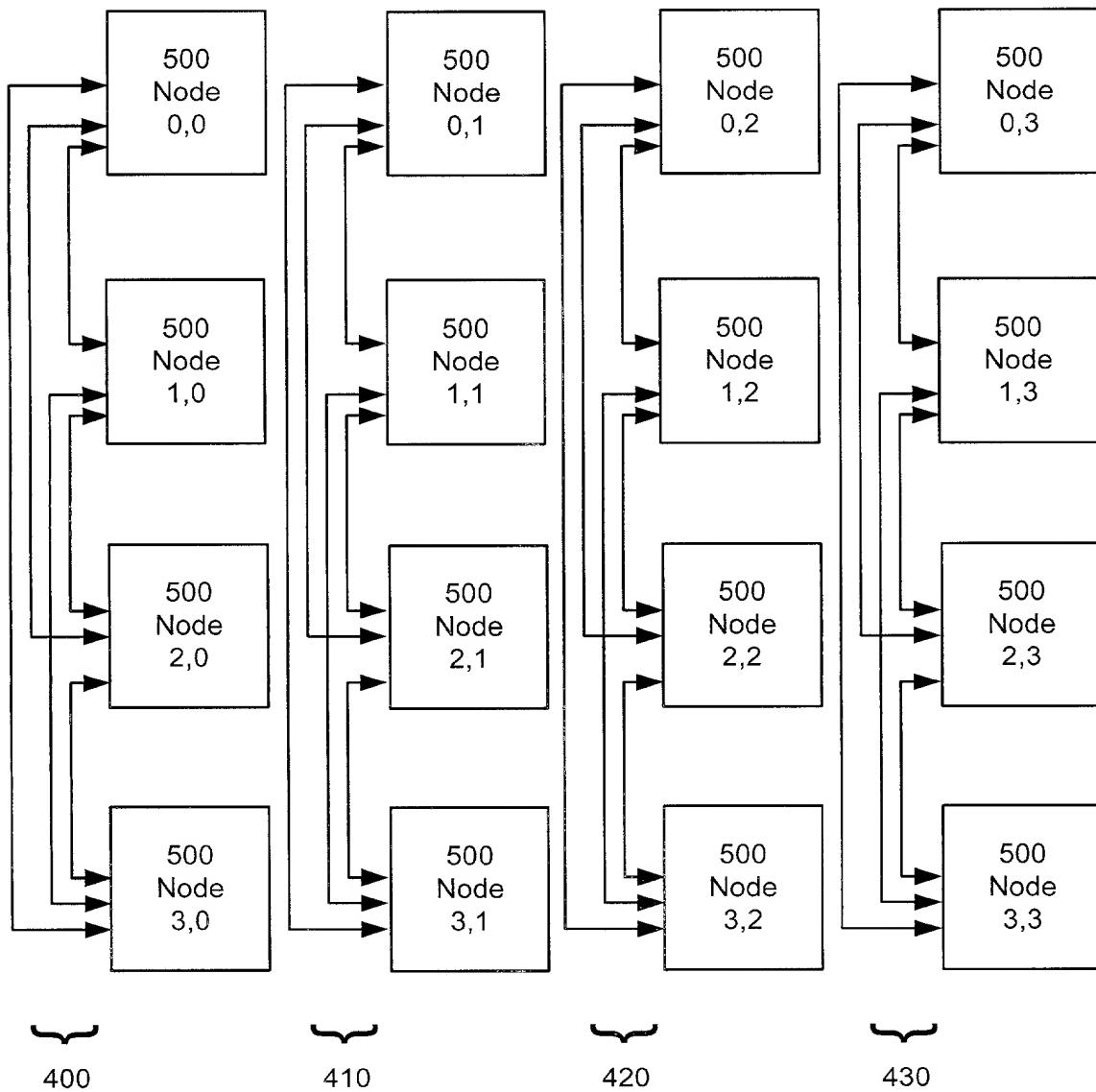


Fig. 7

AGLE0003

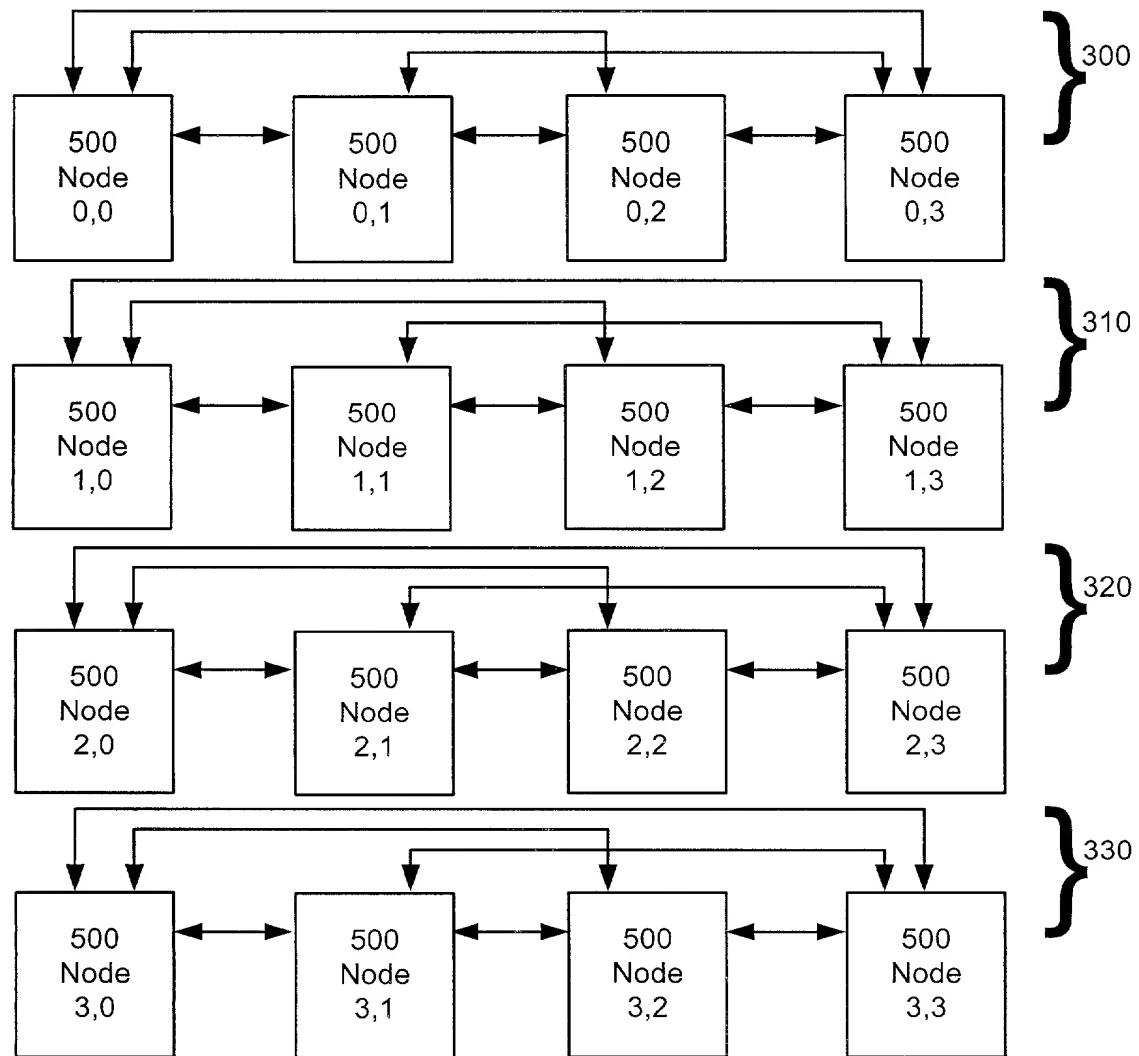


Fig. 8

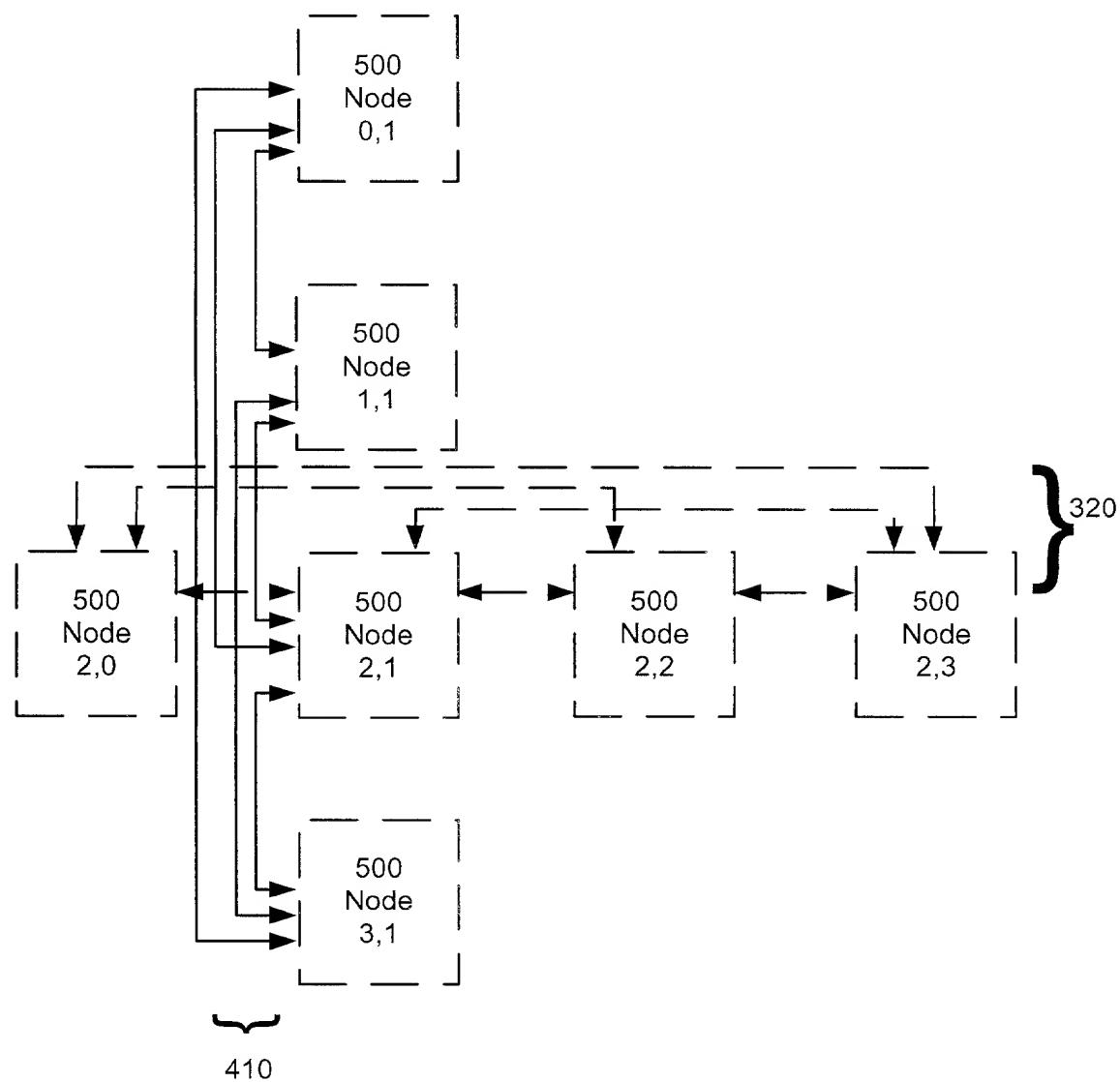


Fig. 9

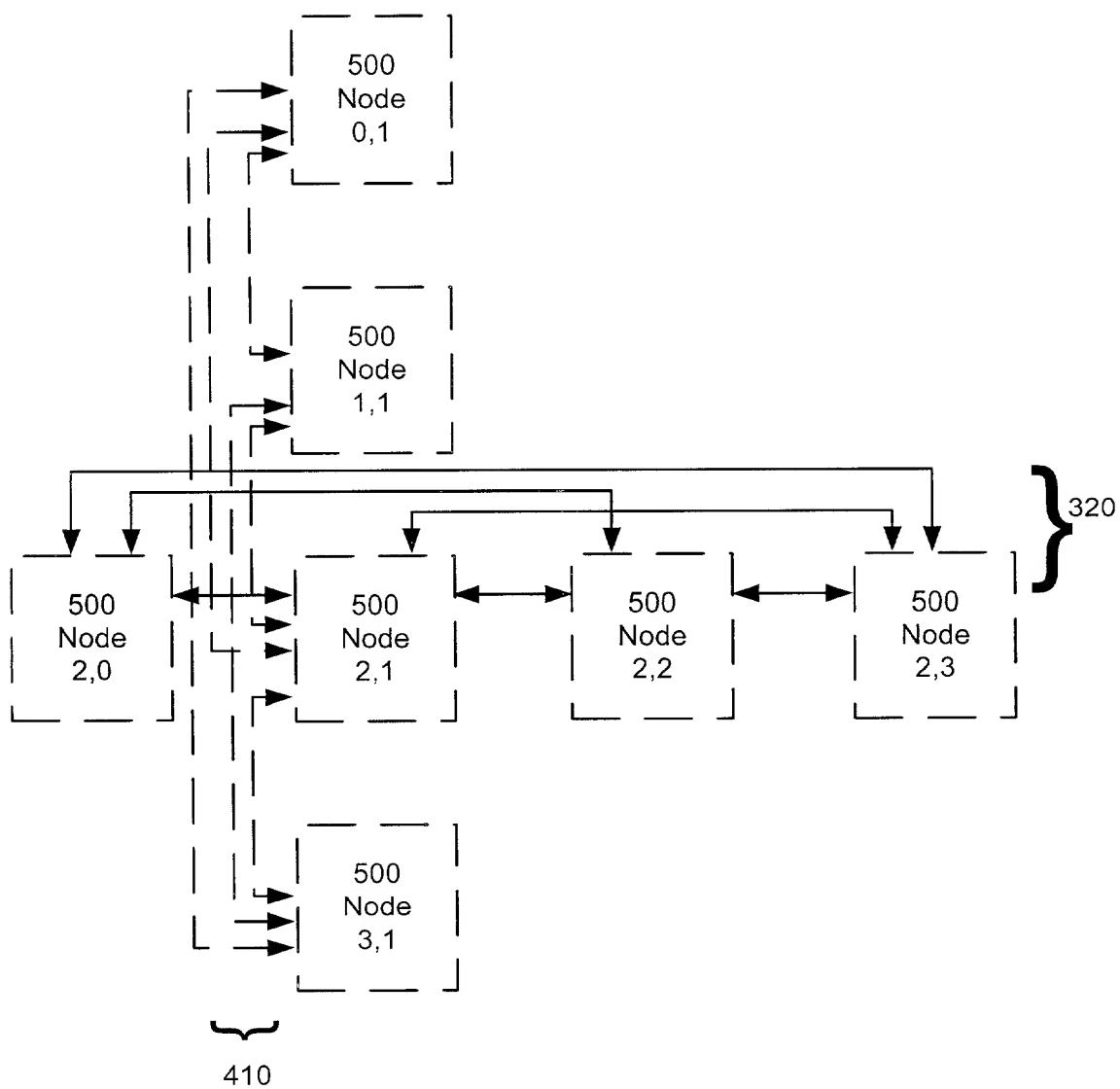


Fig. 10

AGLE0003

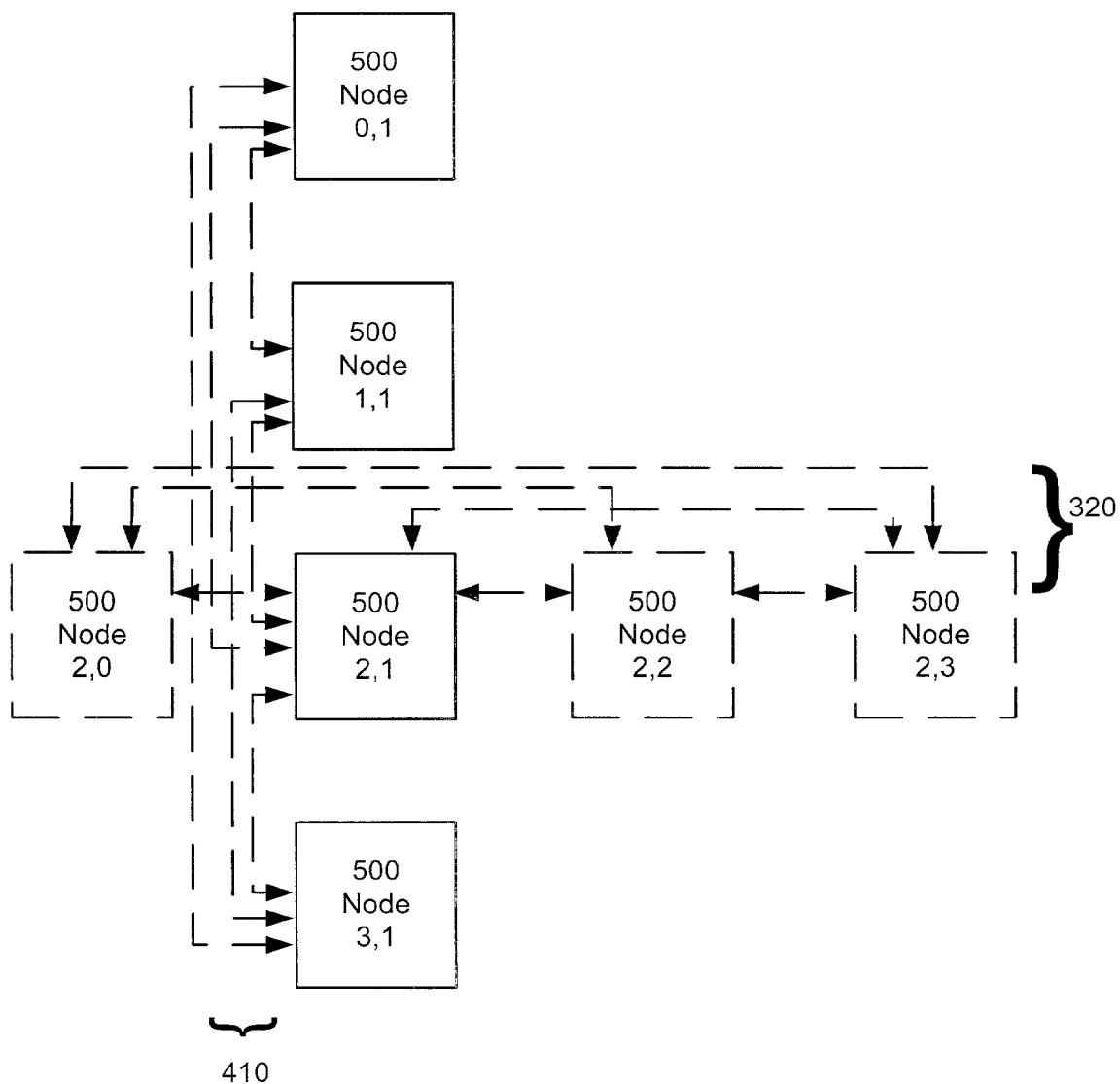


Fig. 11

AGLE0003

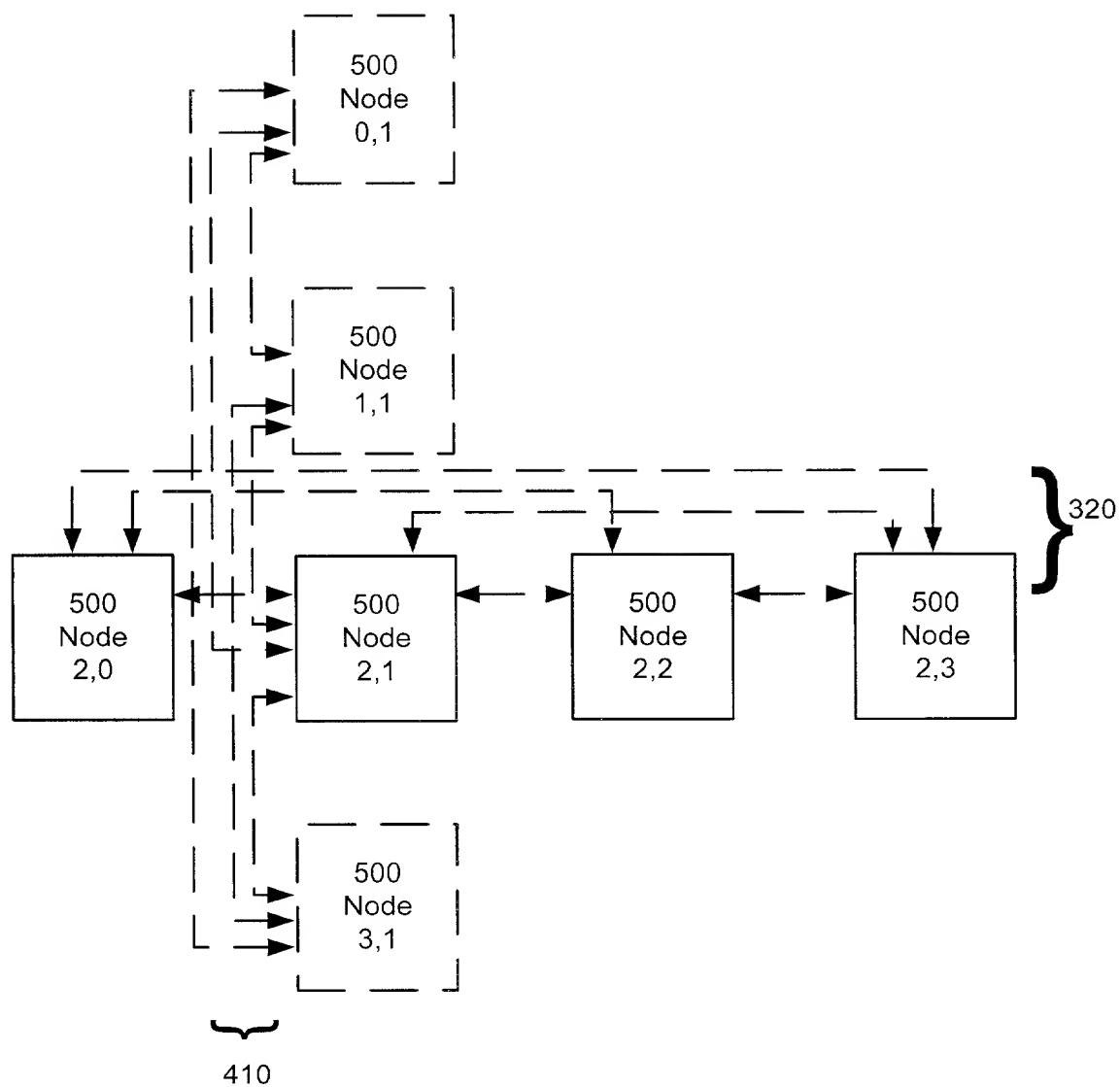


Fig. 12

AGLE0003

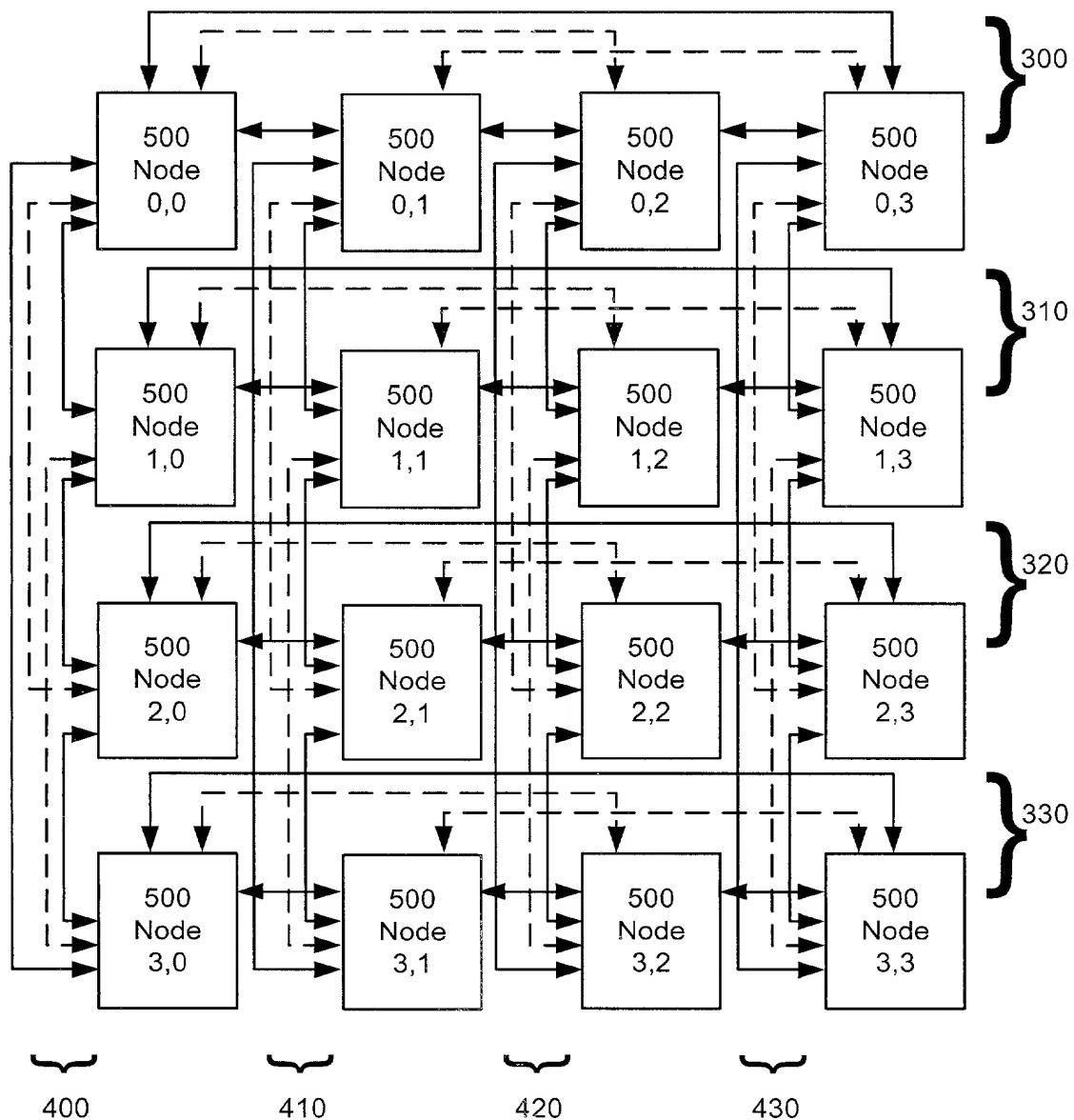


Fig. 13A

AGLE0003

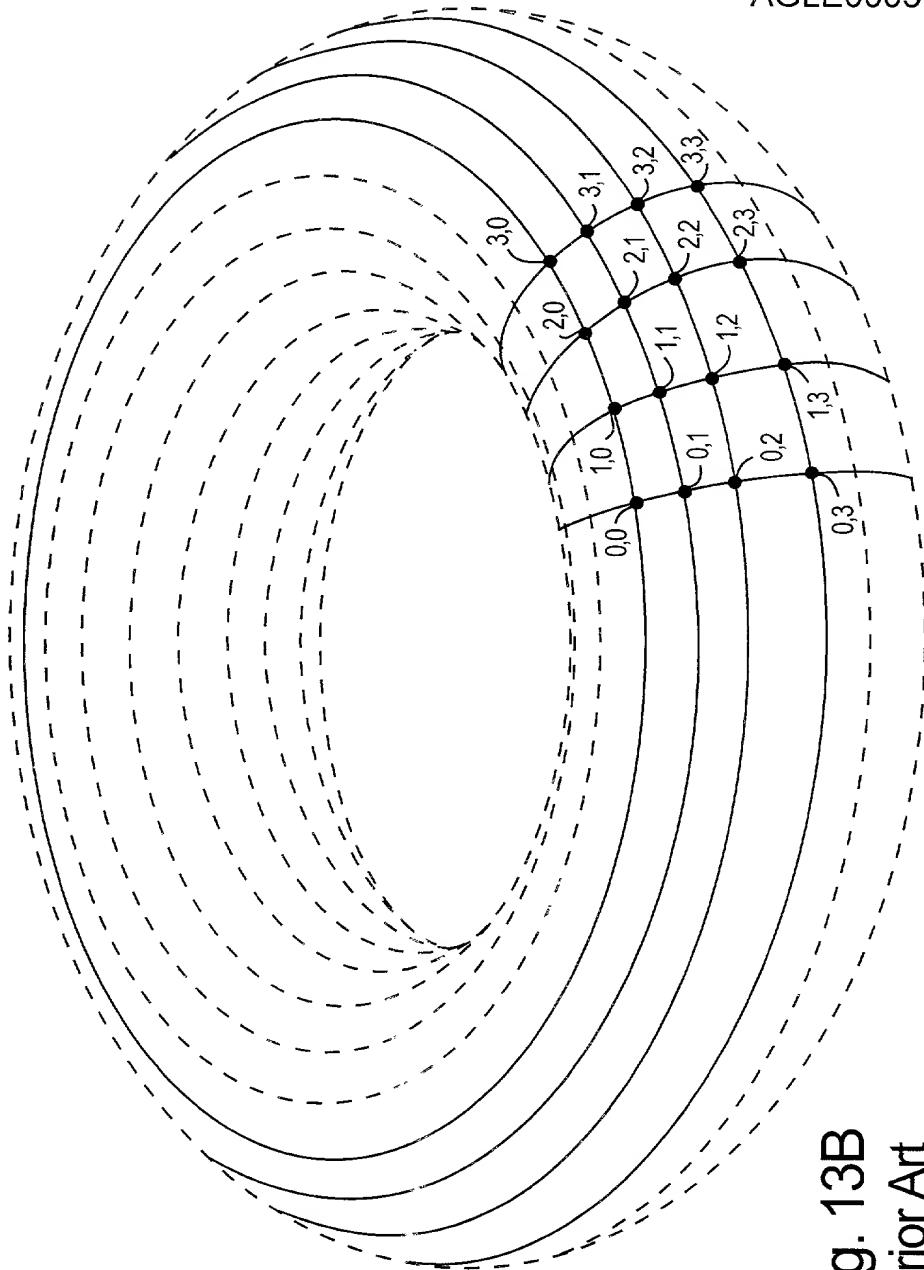


Fig. 13B
Prior Art

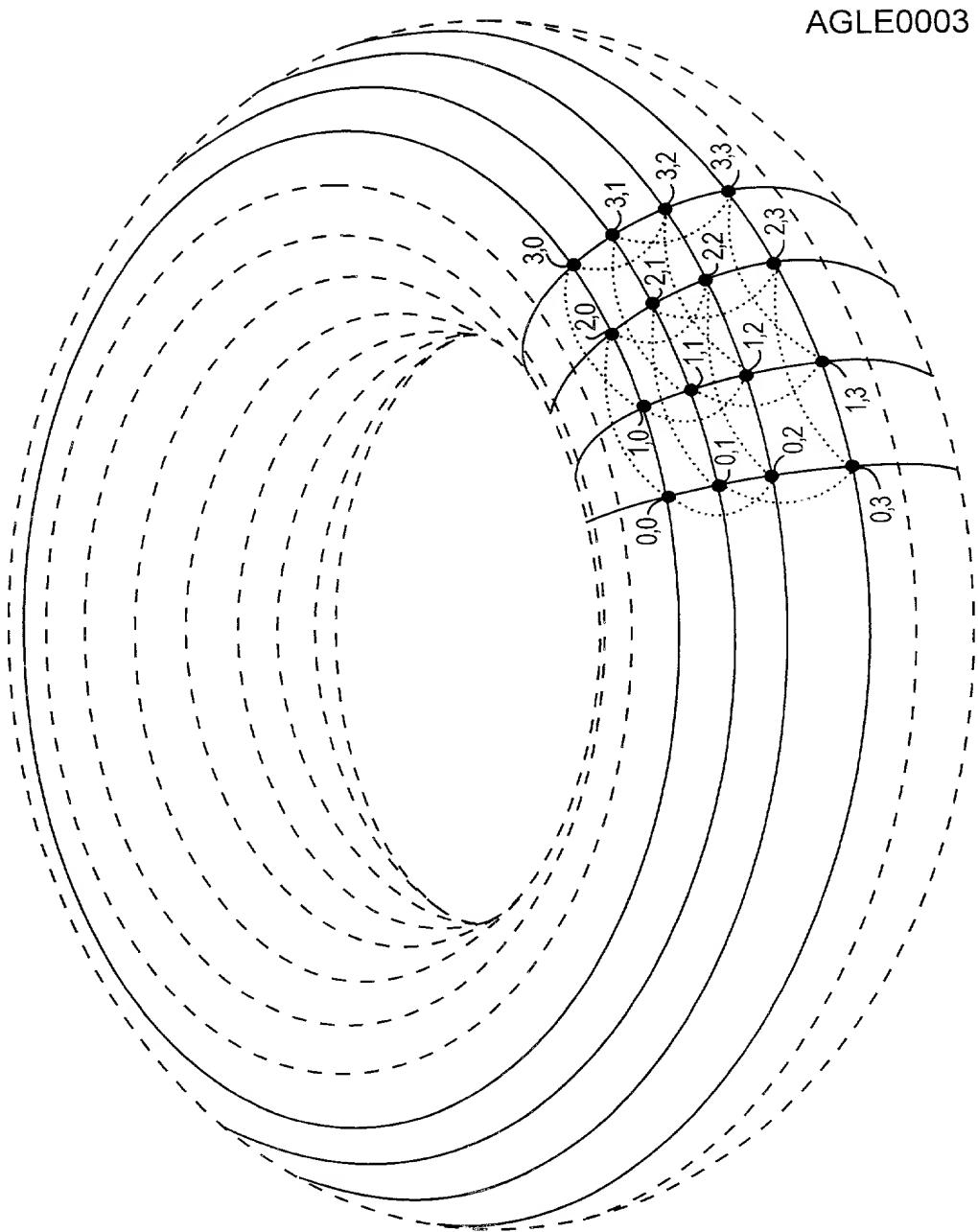


Fig. 13C

AGLE0003

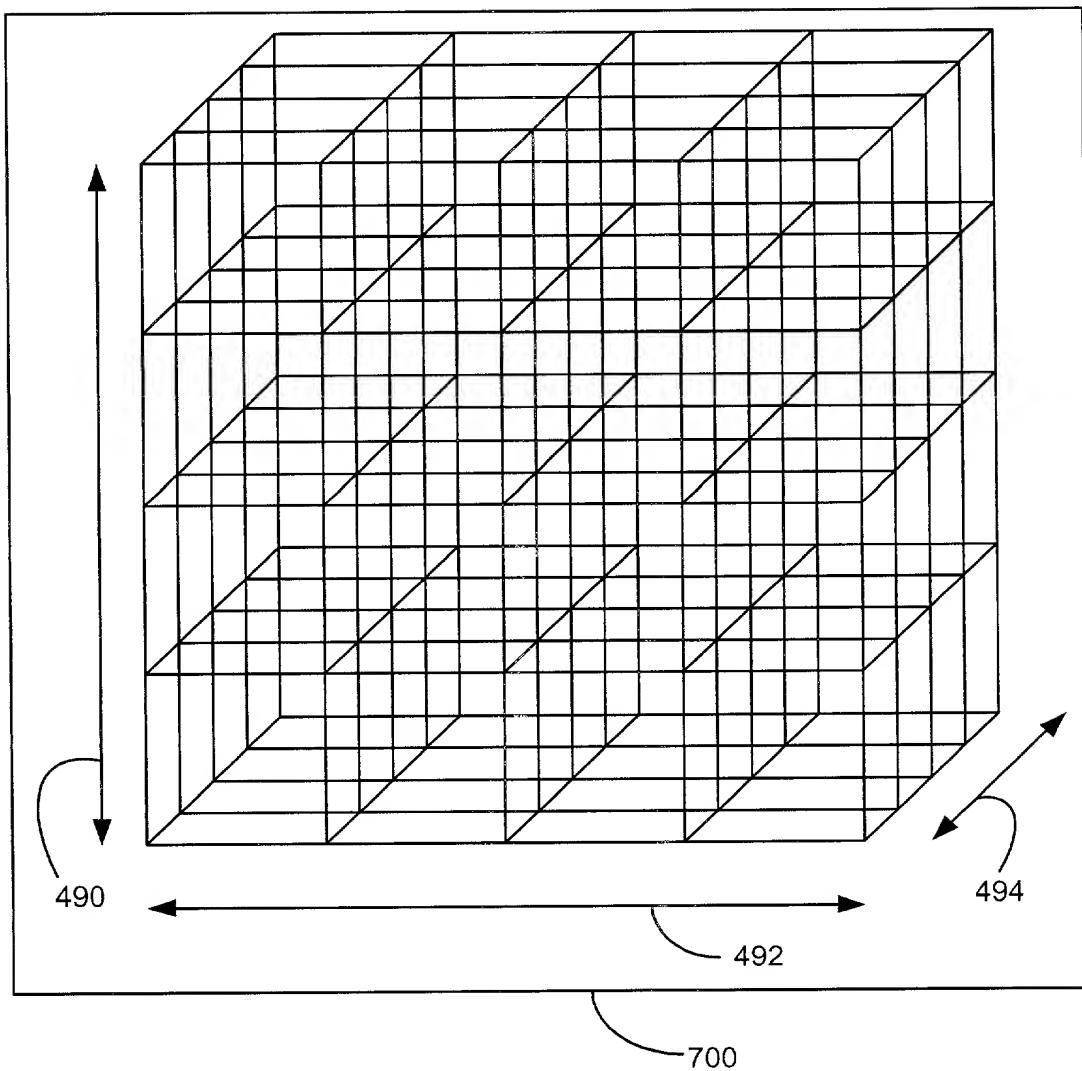


Fig. 14

AGLE0003

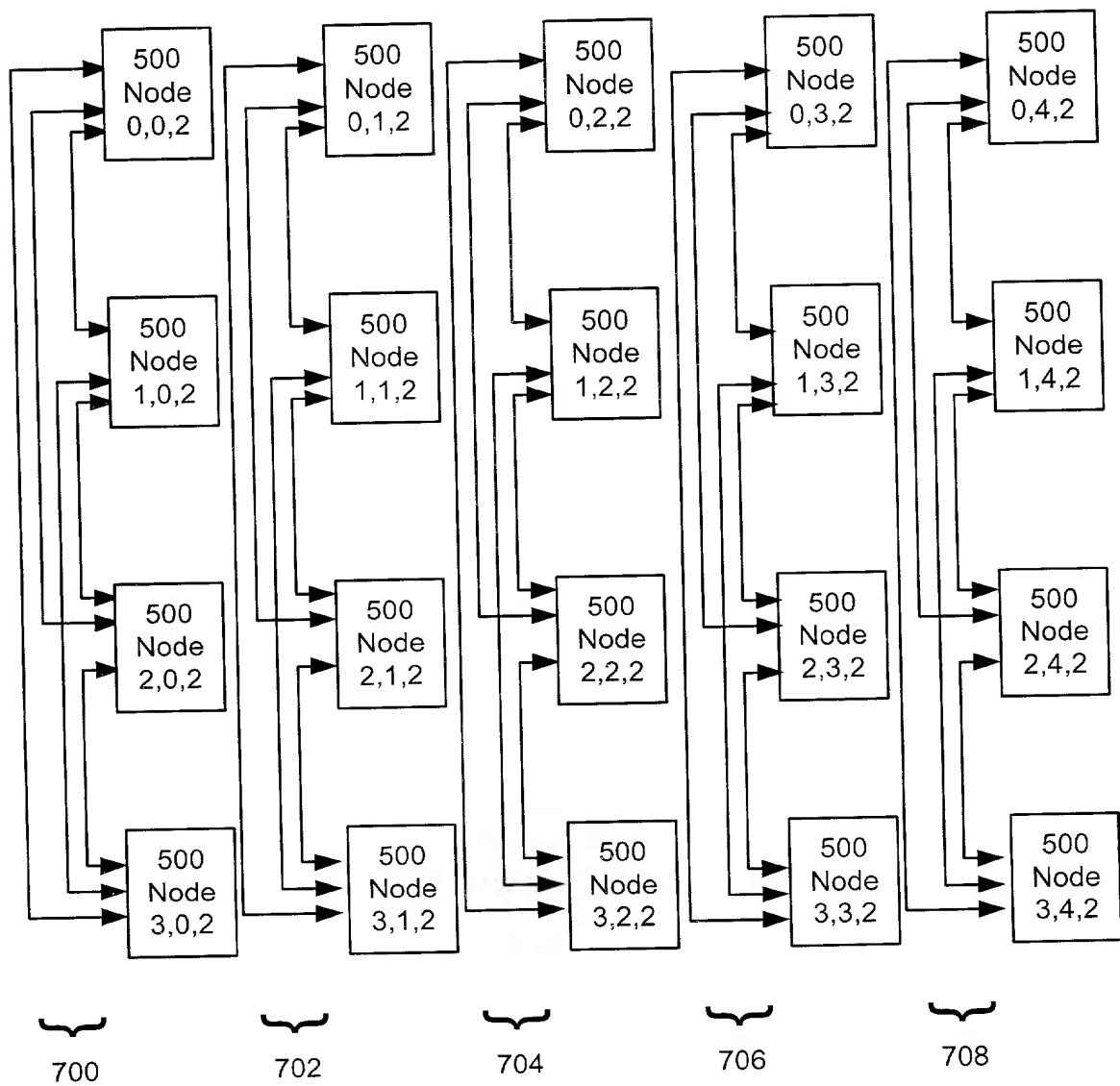


Fig. 15

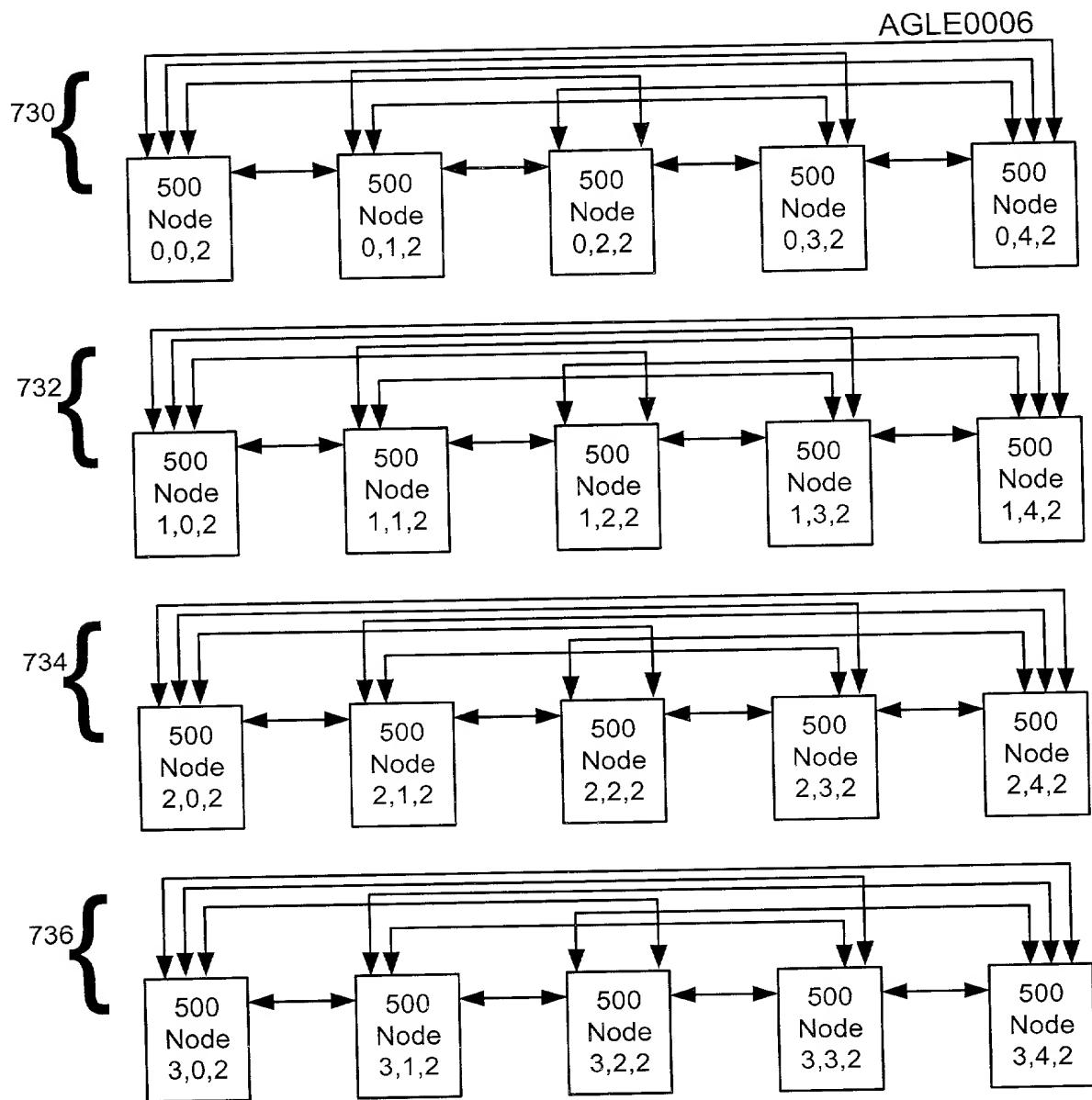


Fig. 16

AGLE0003

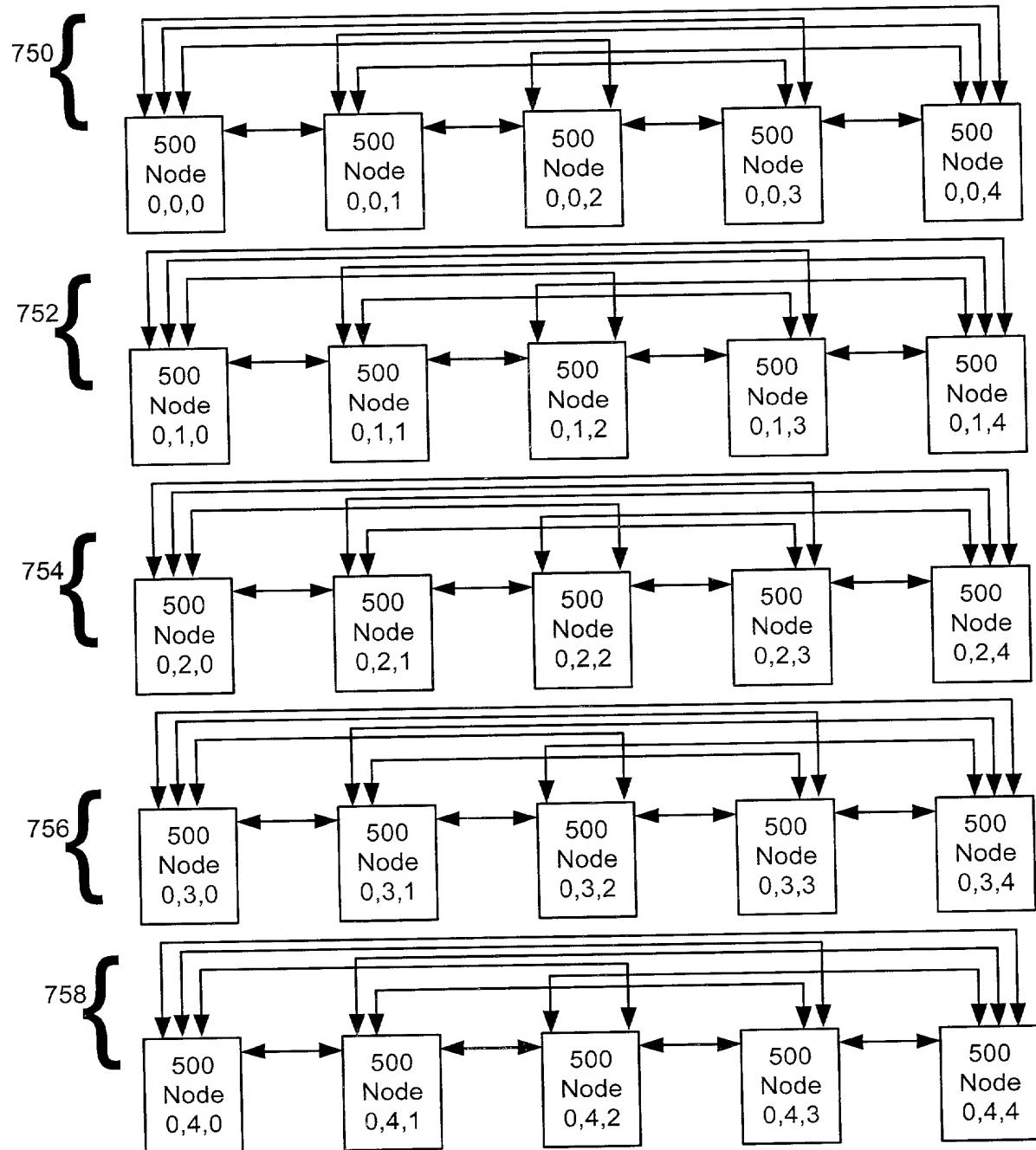


Fig. 17

AGLE0003

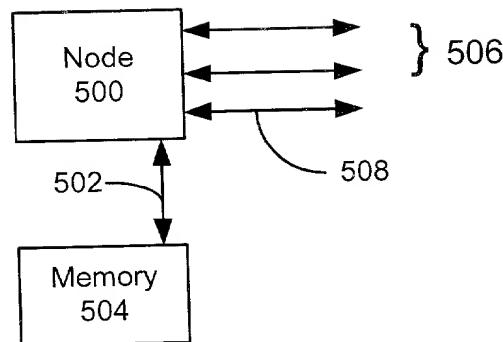


Fig. 18A

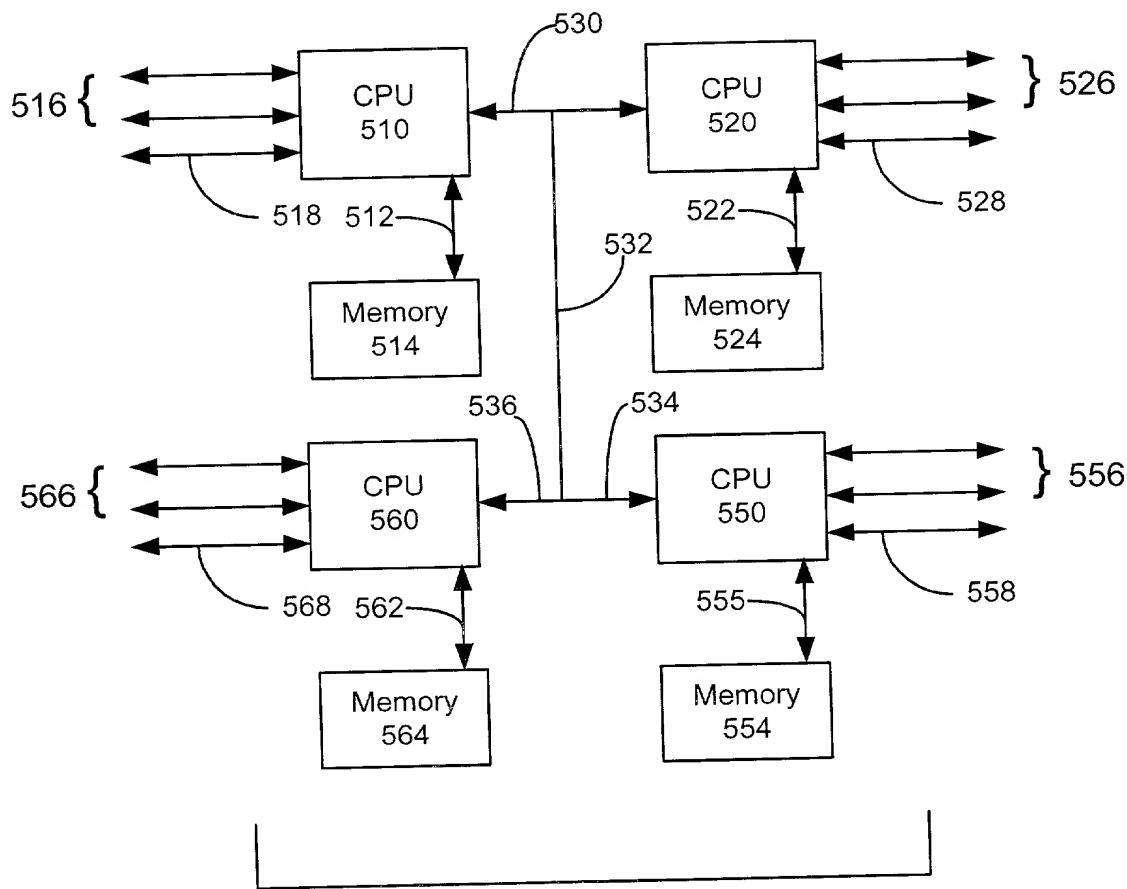


Fig. 18B

AGLE0003

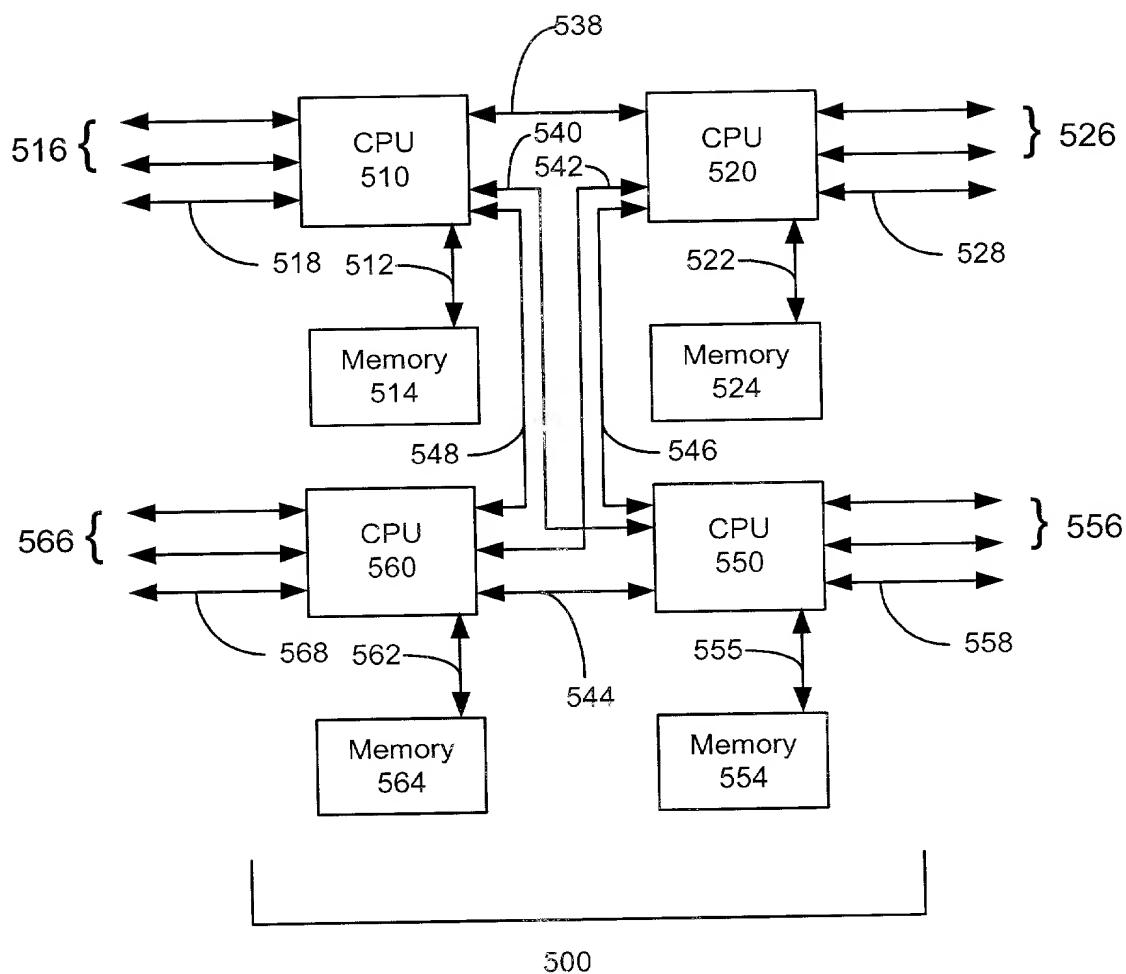


Fig. 19

AGLE0003

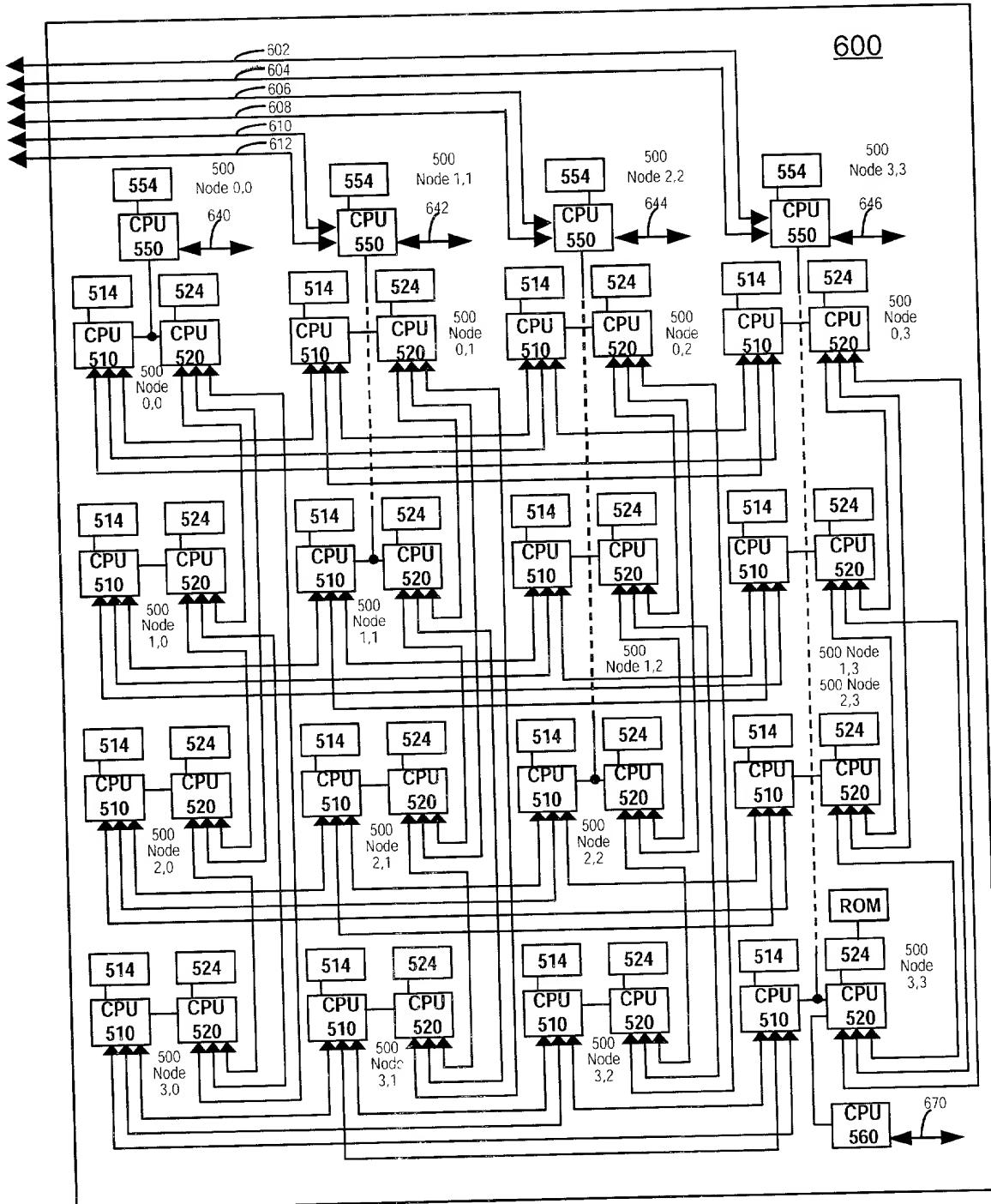


Fig. 20A

AGLE0003

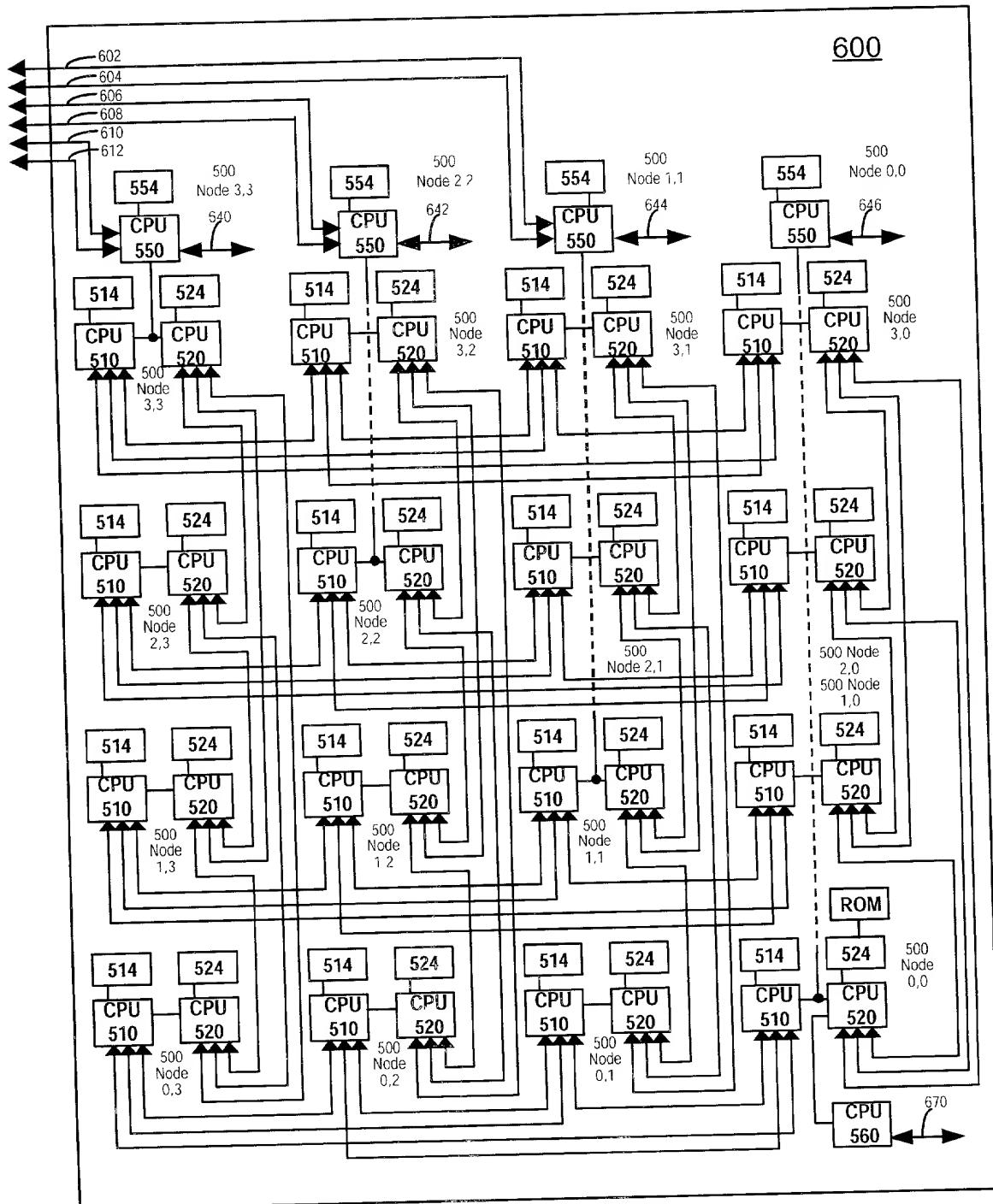


Fig. 20B

AGLE0003

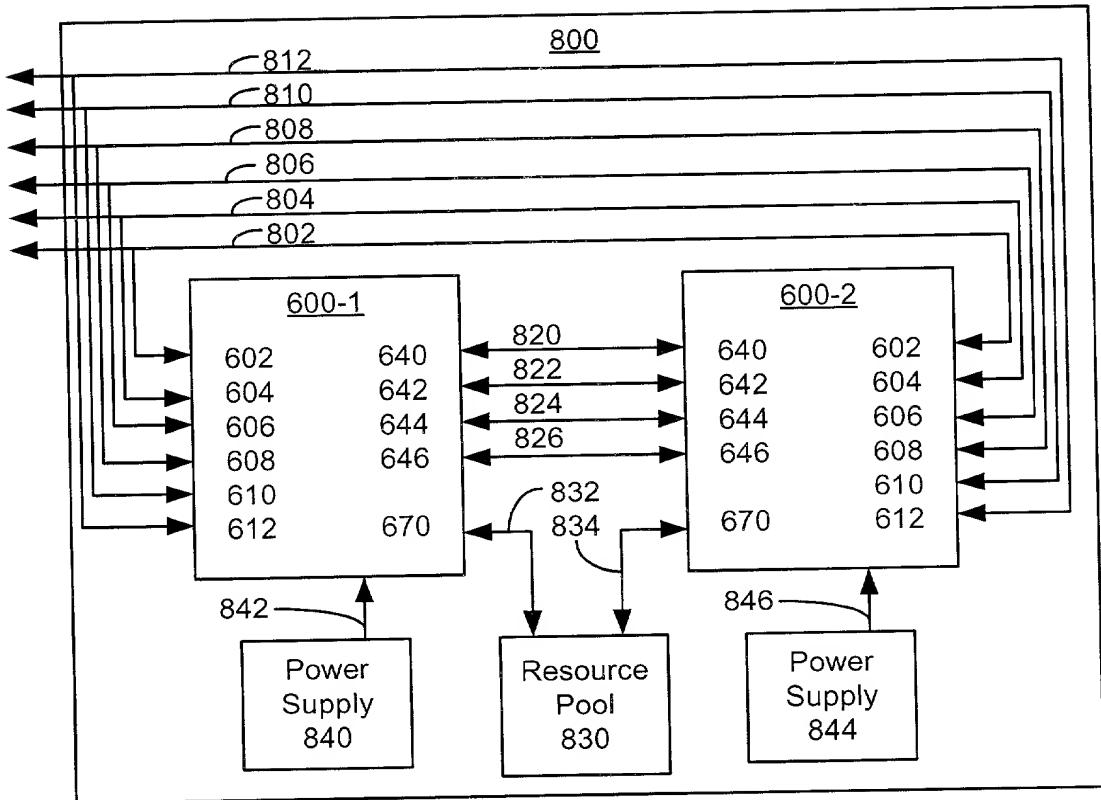


Fig. 21

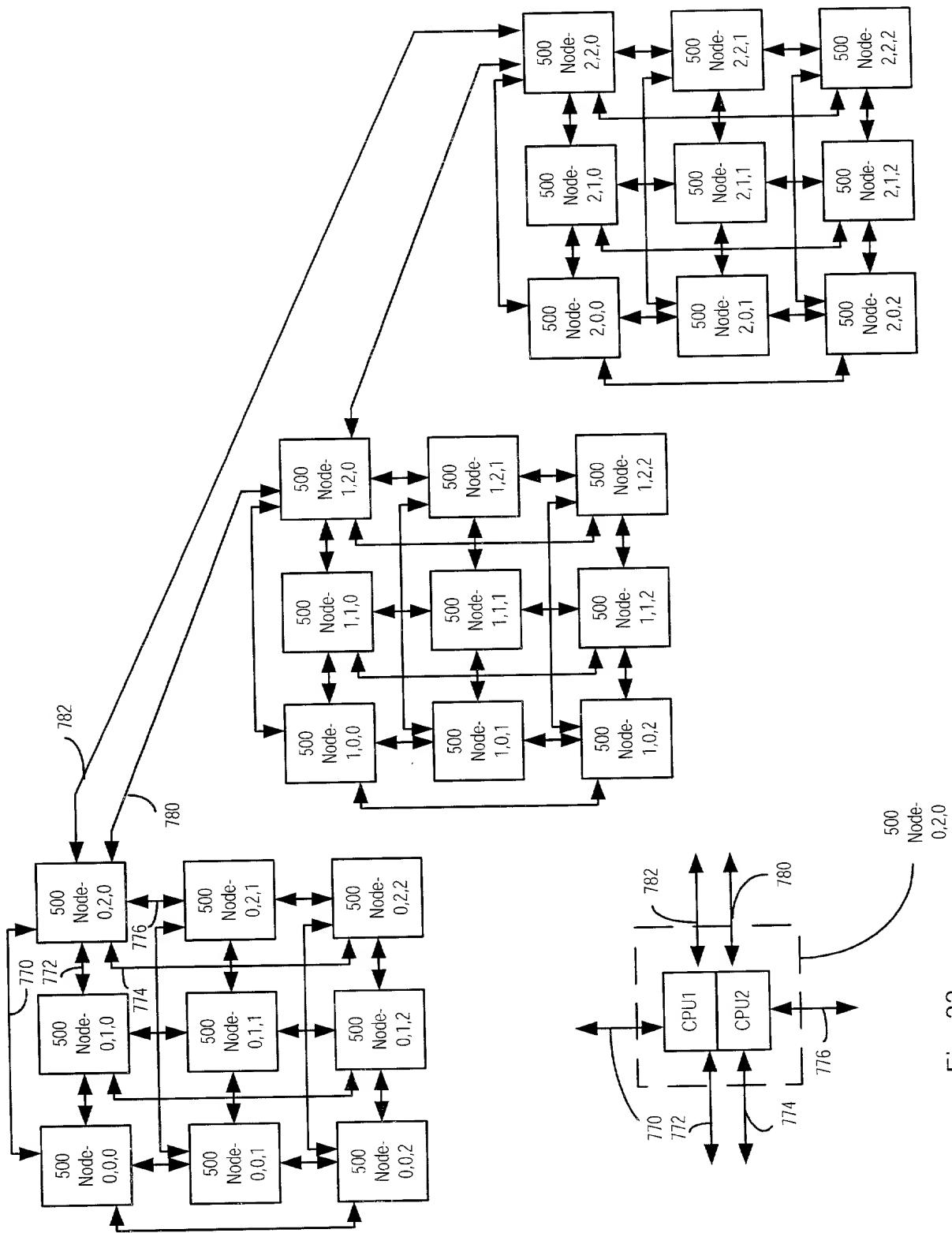
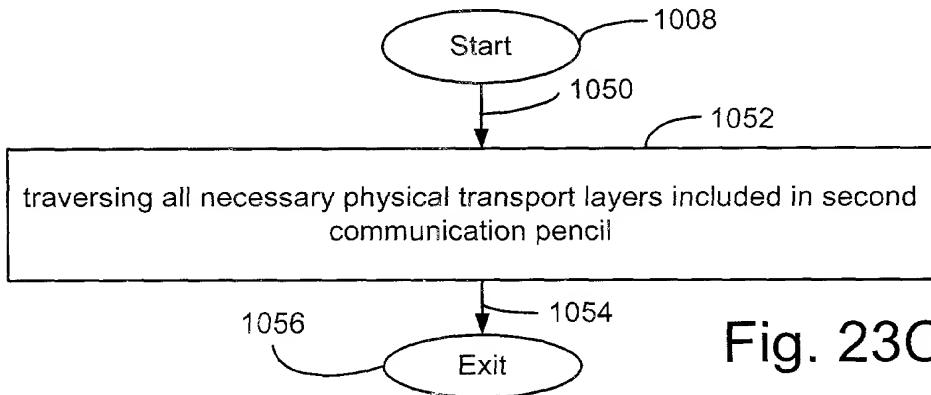
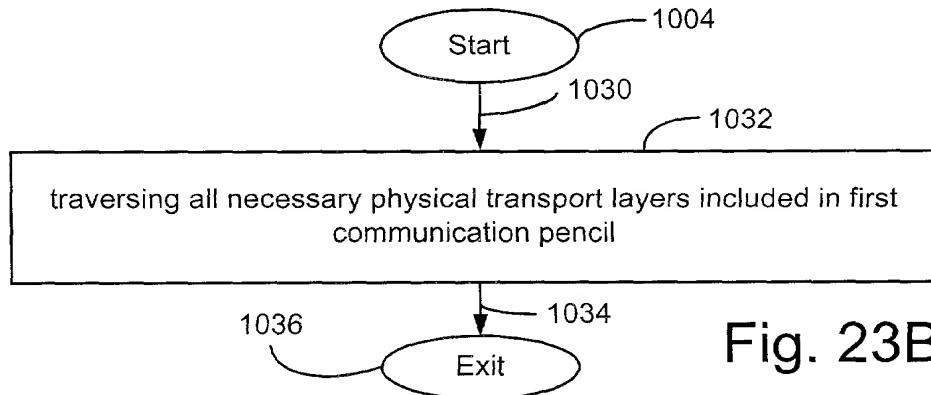
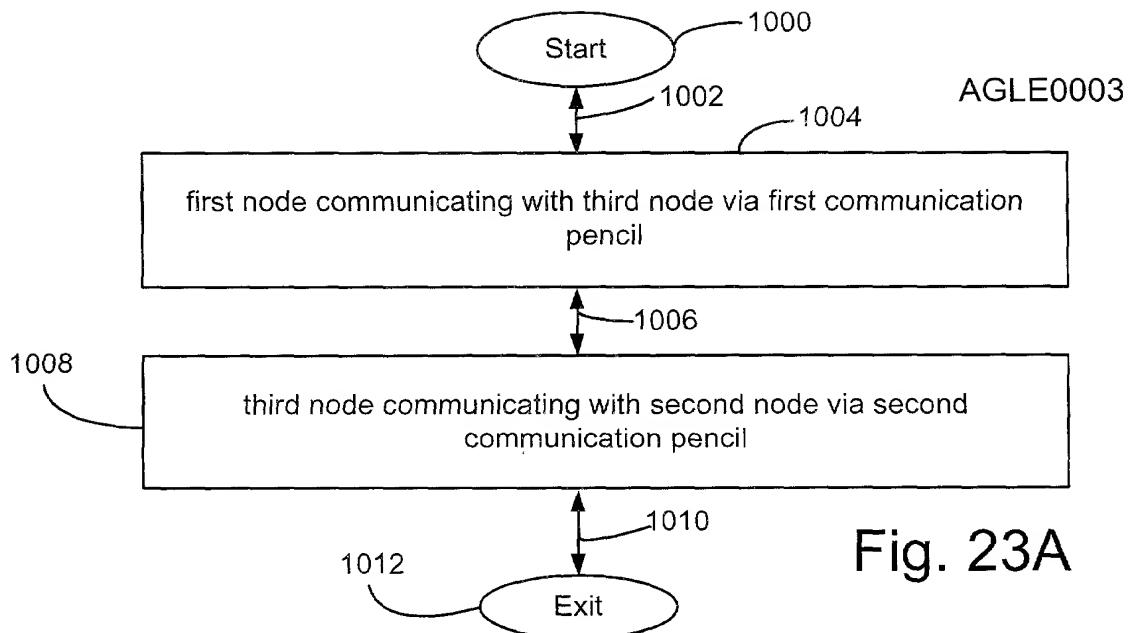


Fig. 22



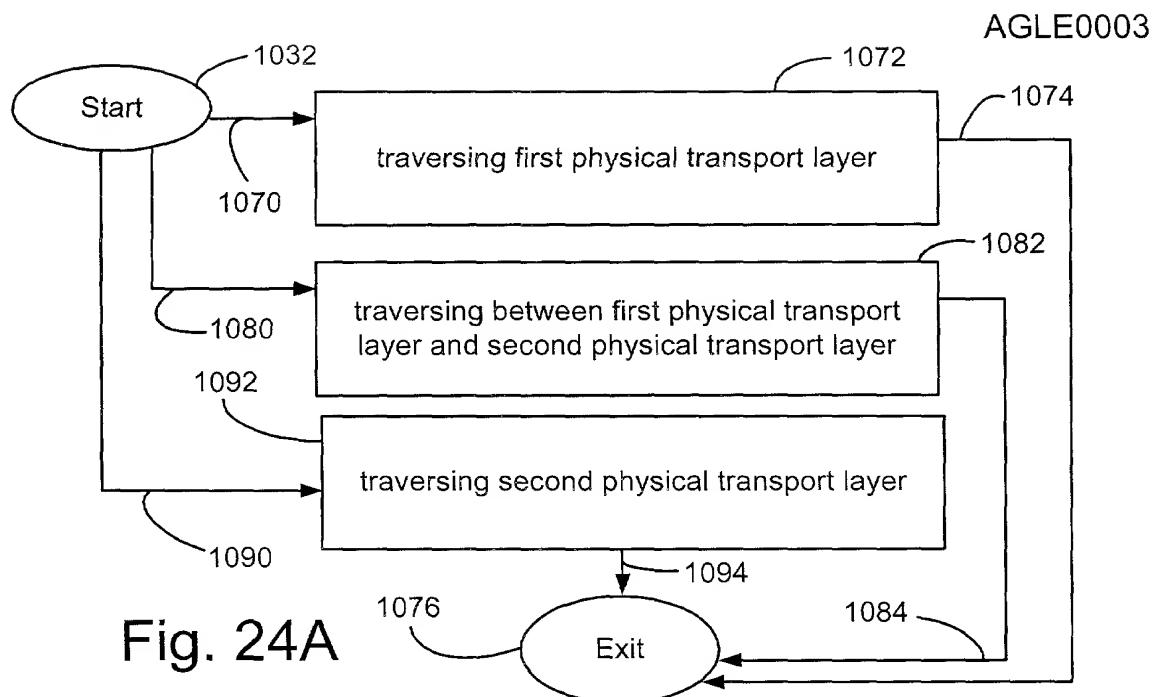


Fig. 24A

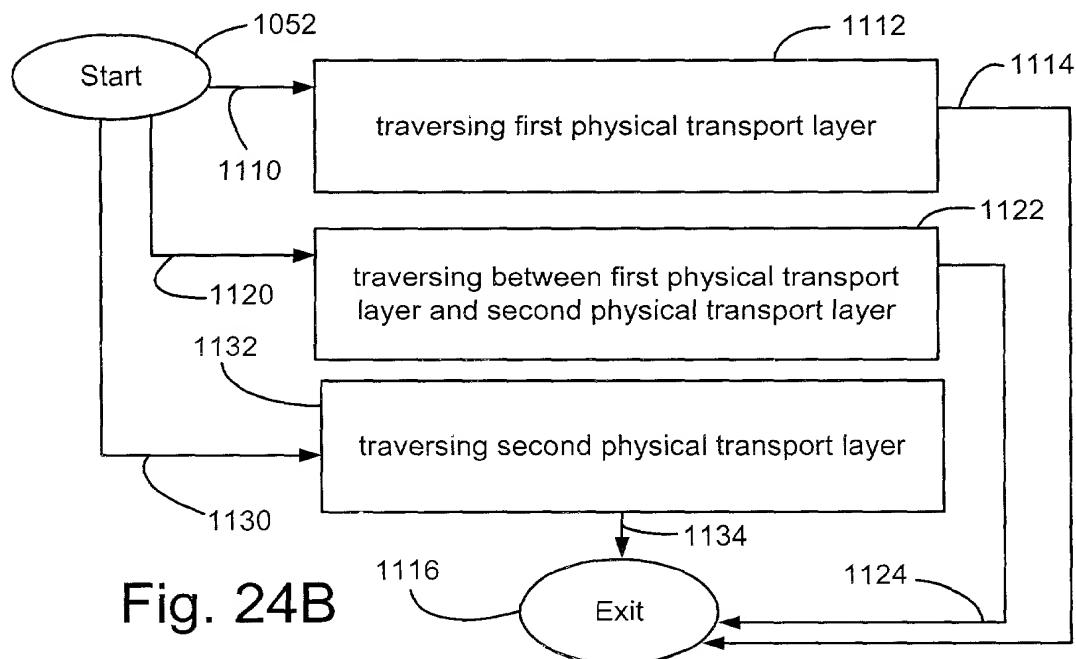


Fig. 24B

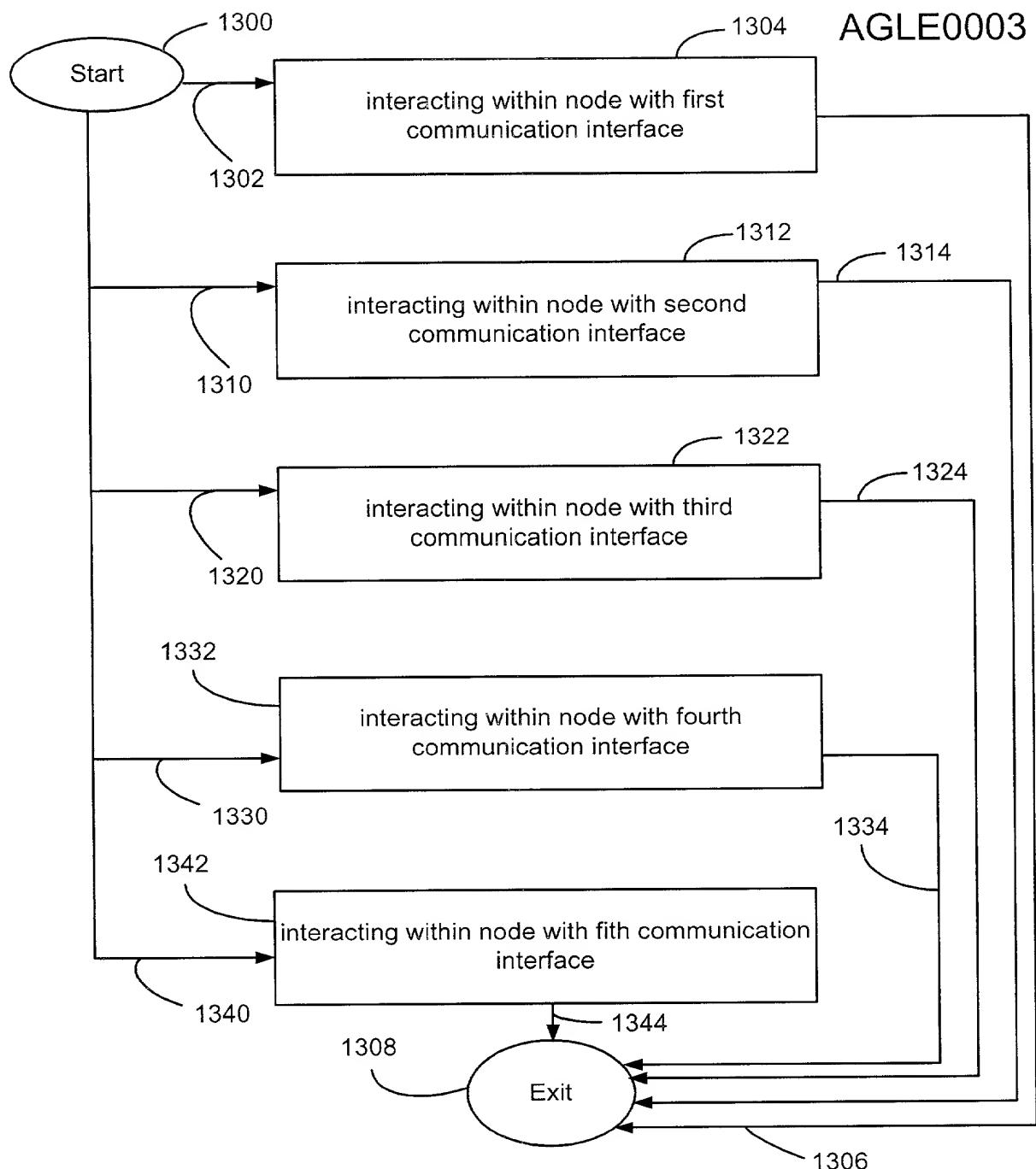


Fig. 25

AGLE0003

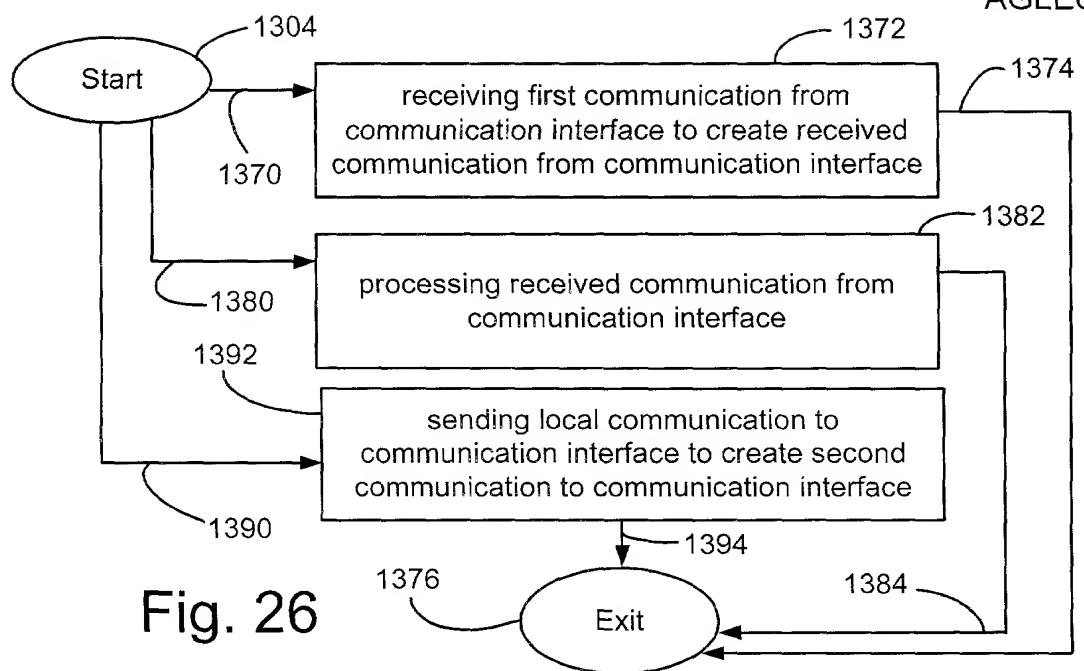
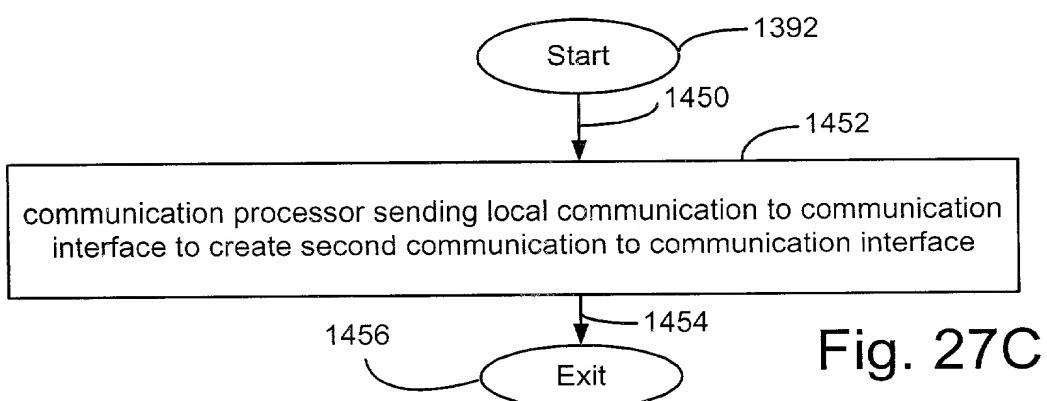
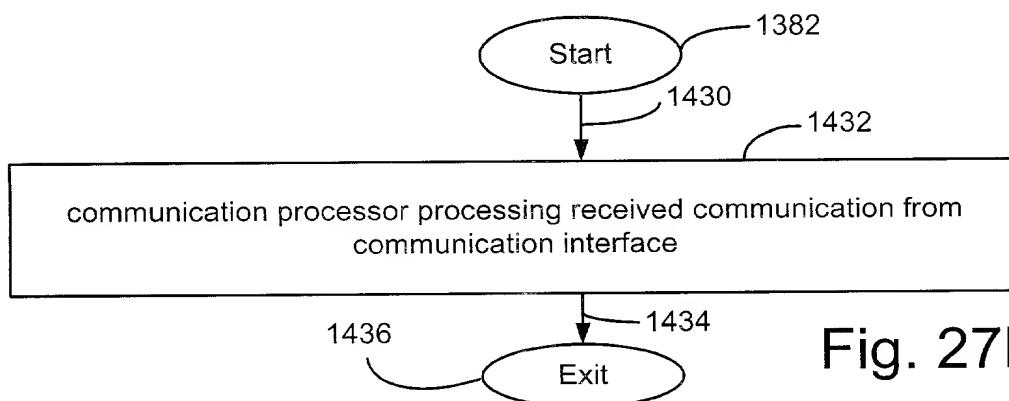
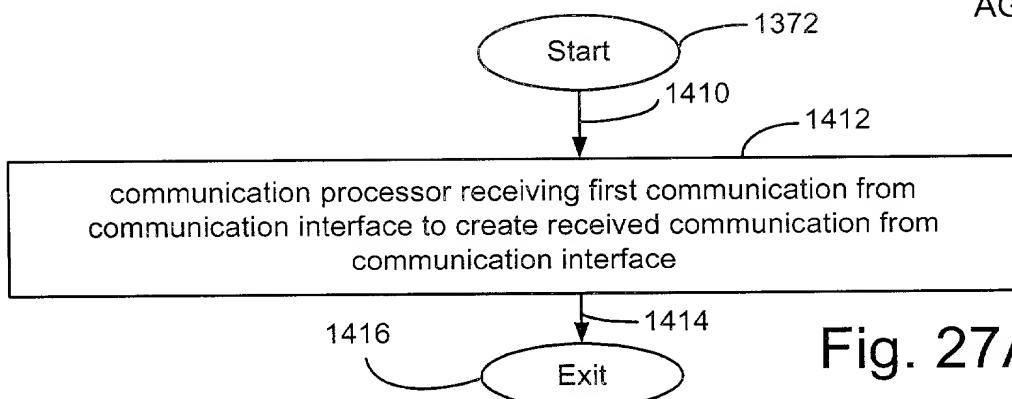


Fig. 26

AGLE0003



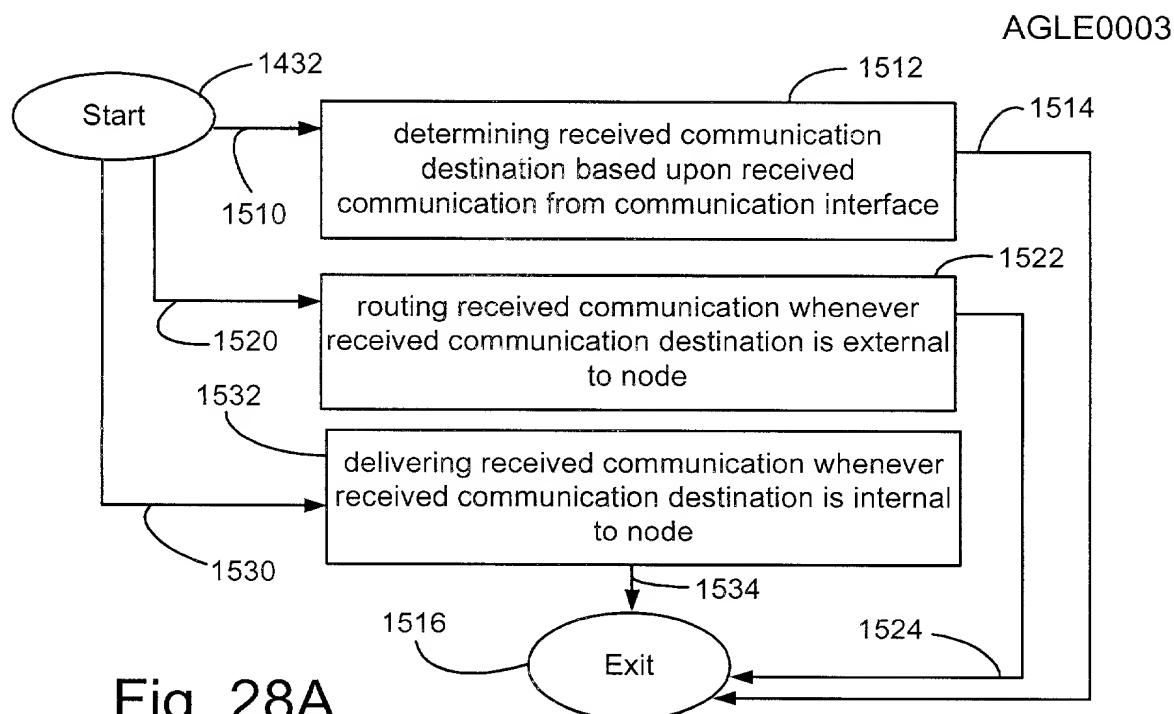


Fig. 28A

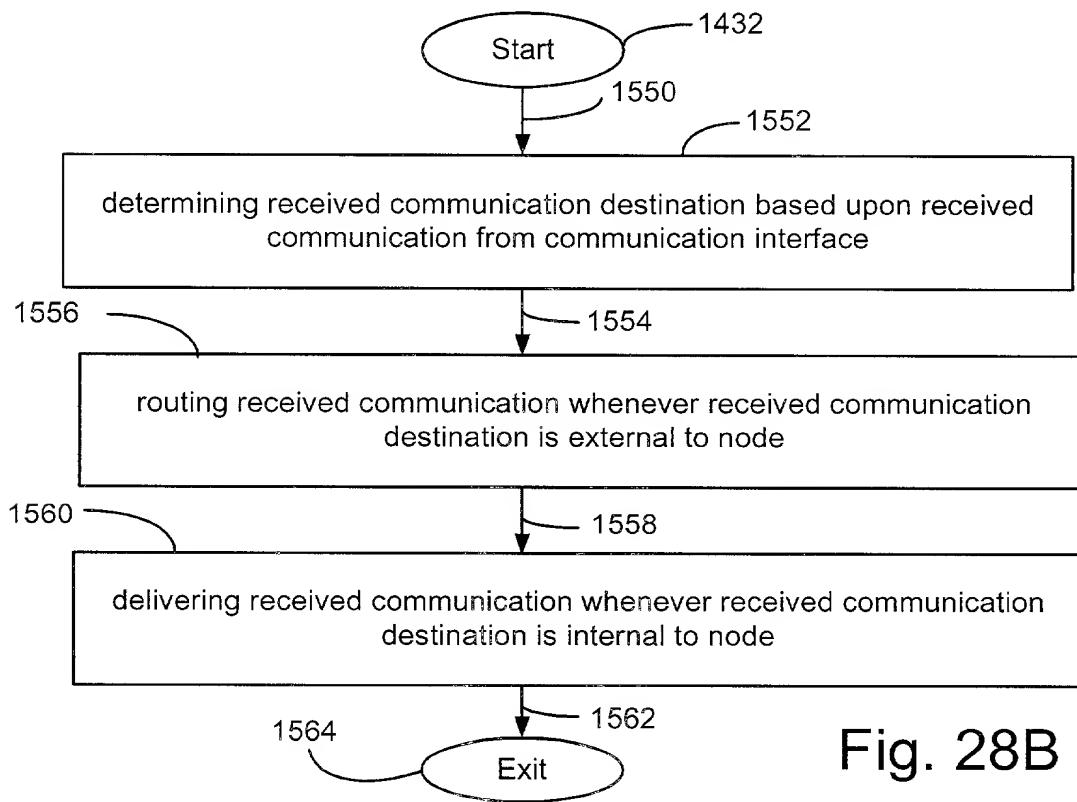


Fig. 28B

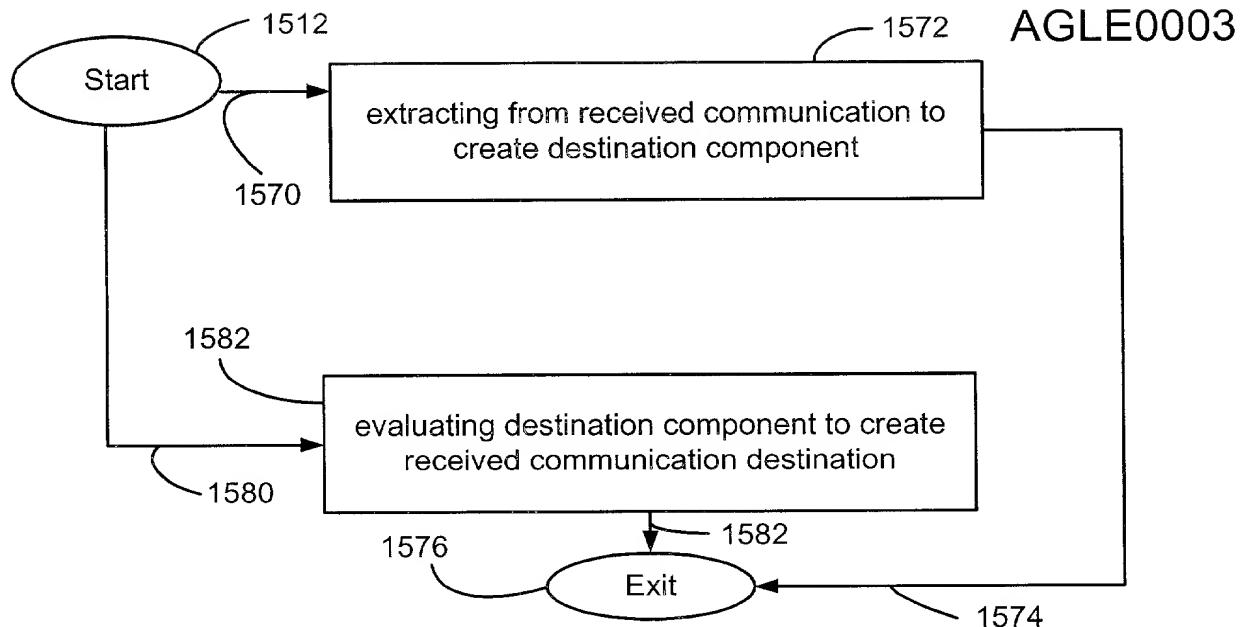


Fig. 29A

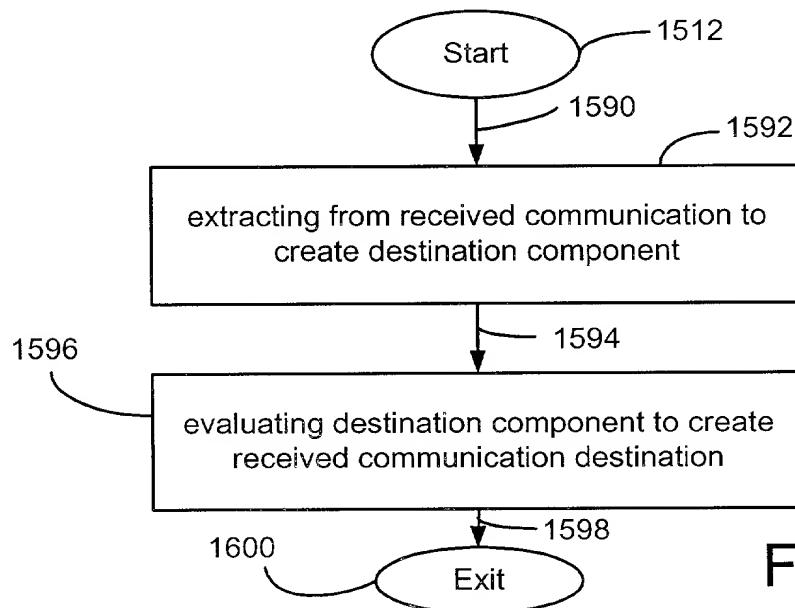


Fig. 29B

AGLE0003

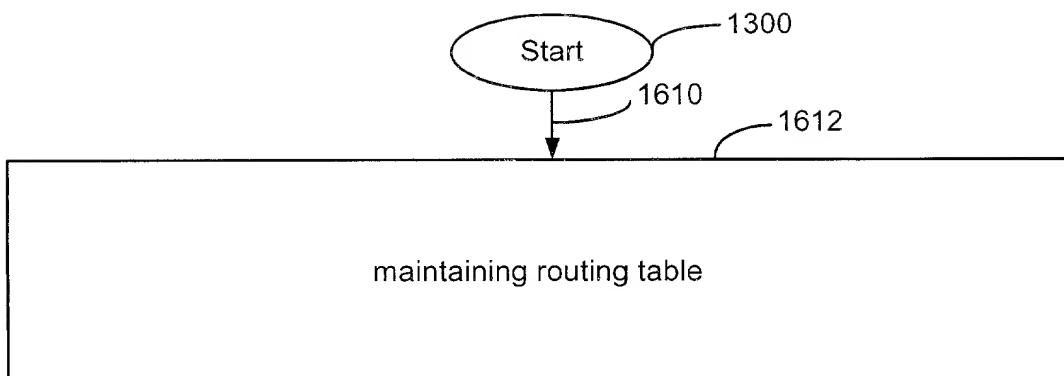


Fig. 30A

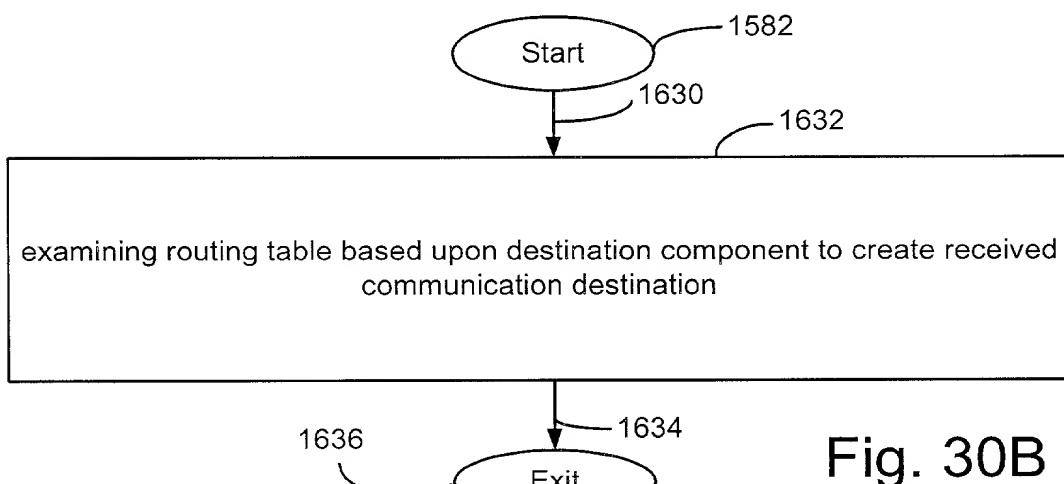
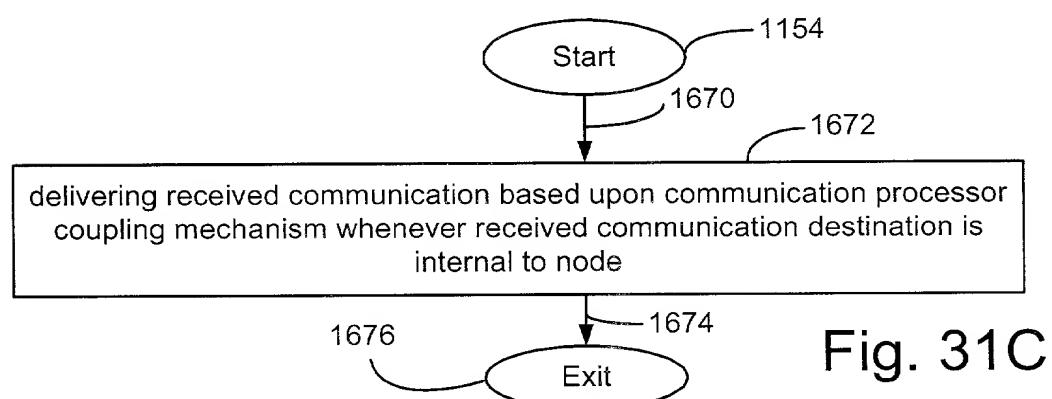
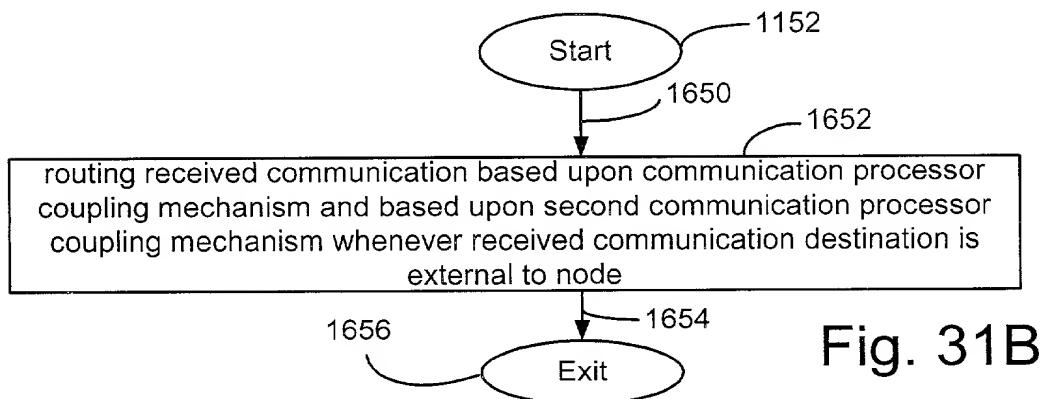
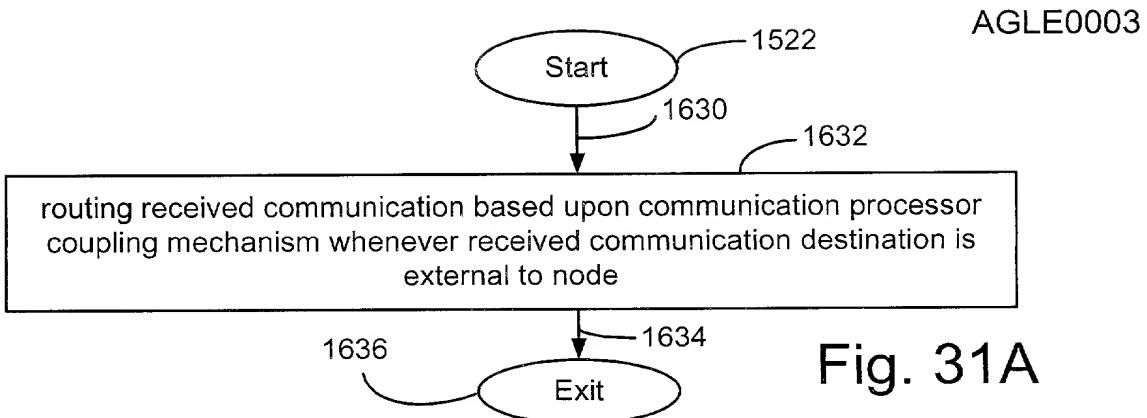
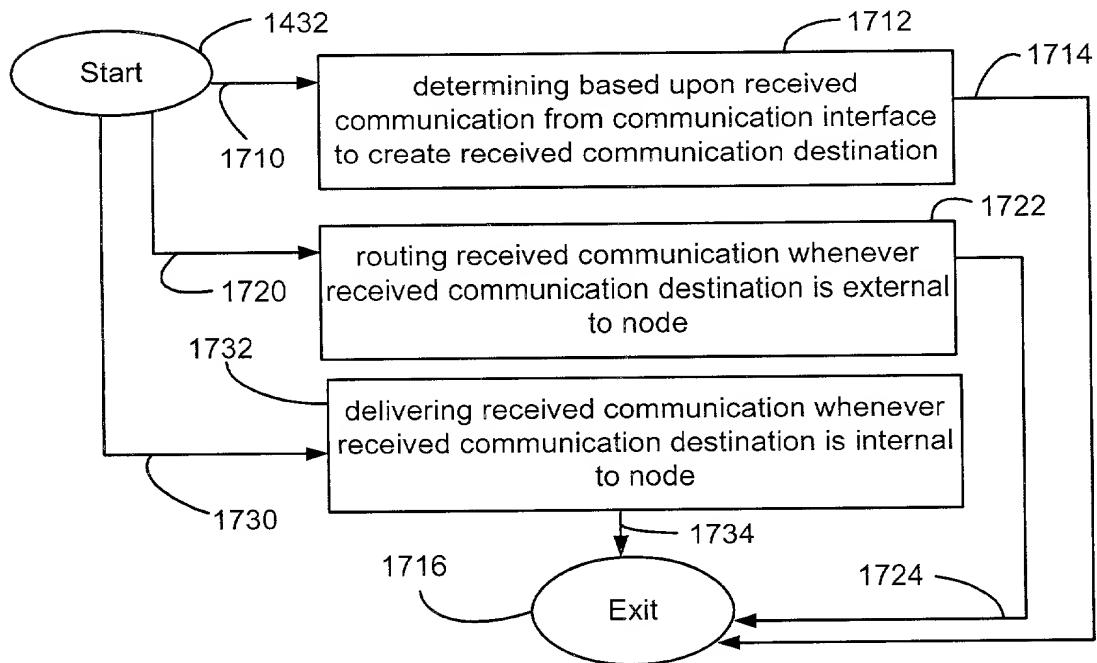
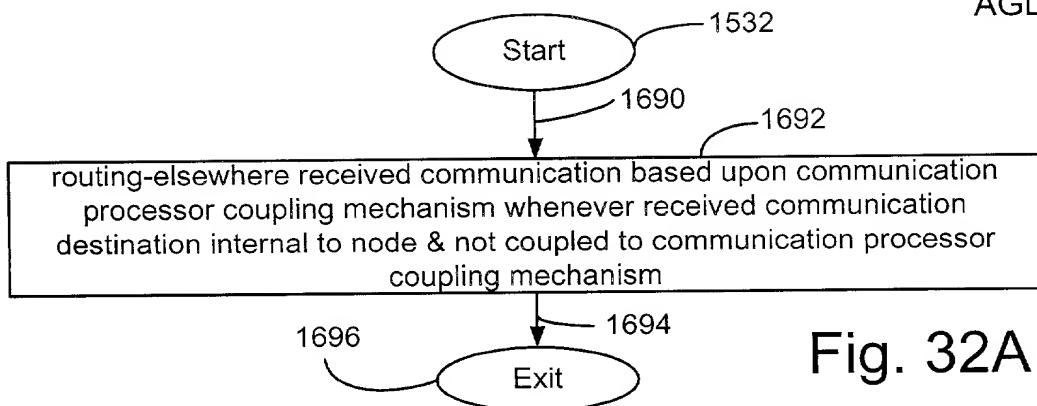


Fig. 30B



AGLE0003



AGLE0003

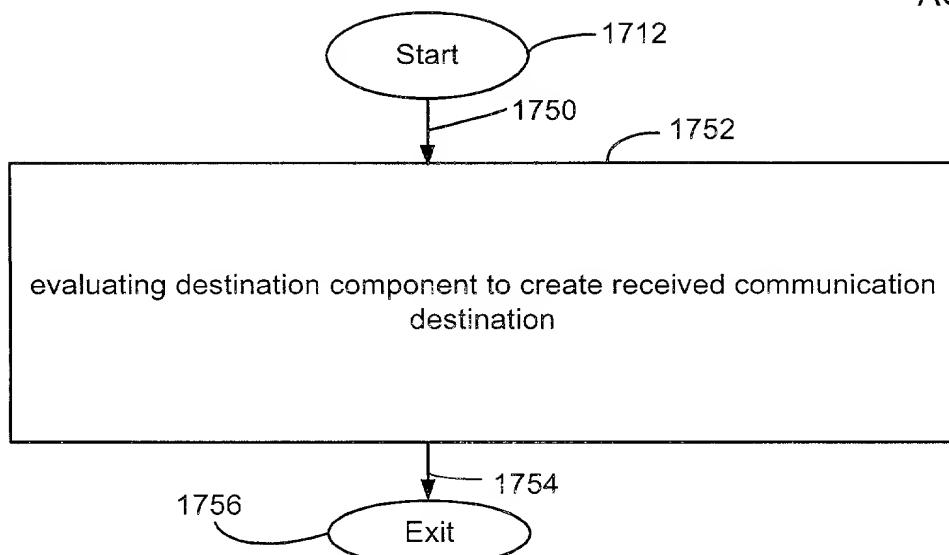


Fig. 33A

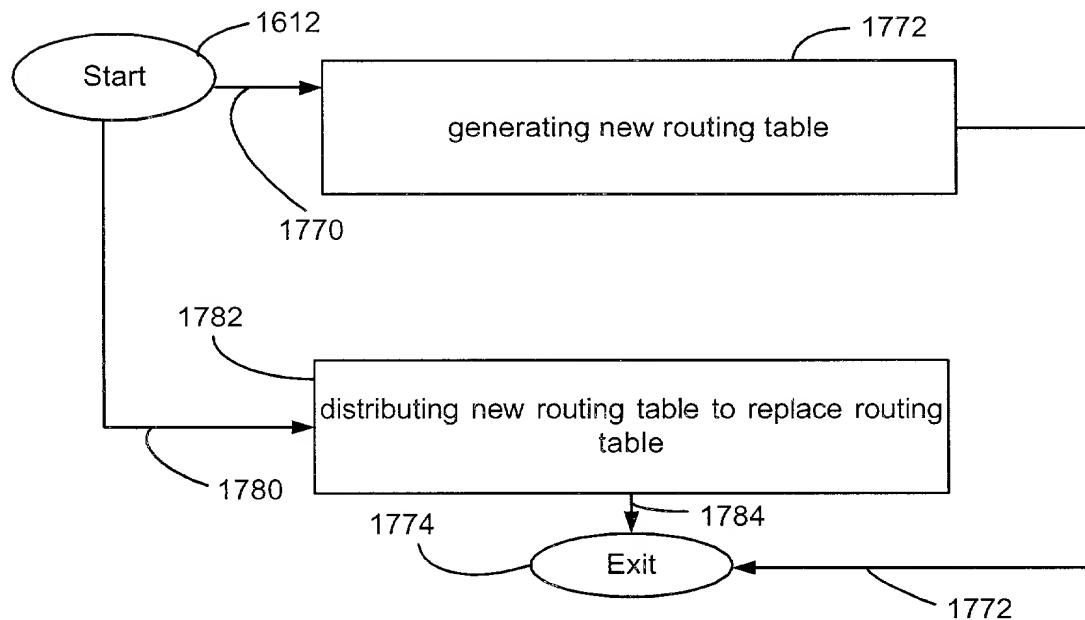
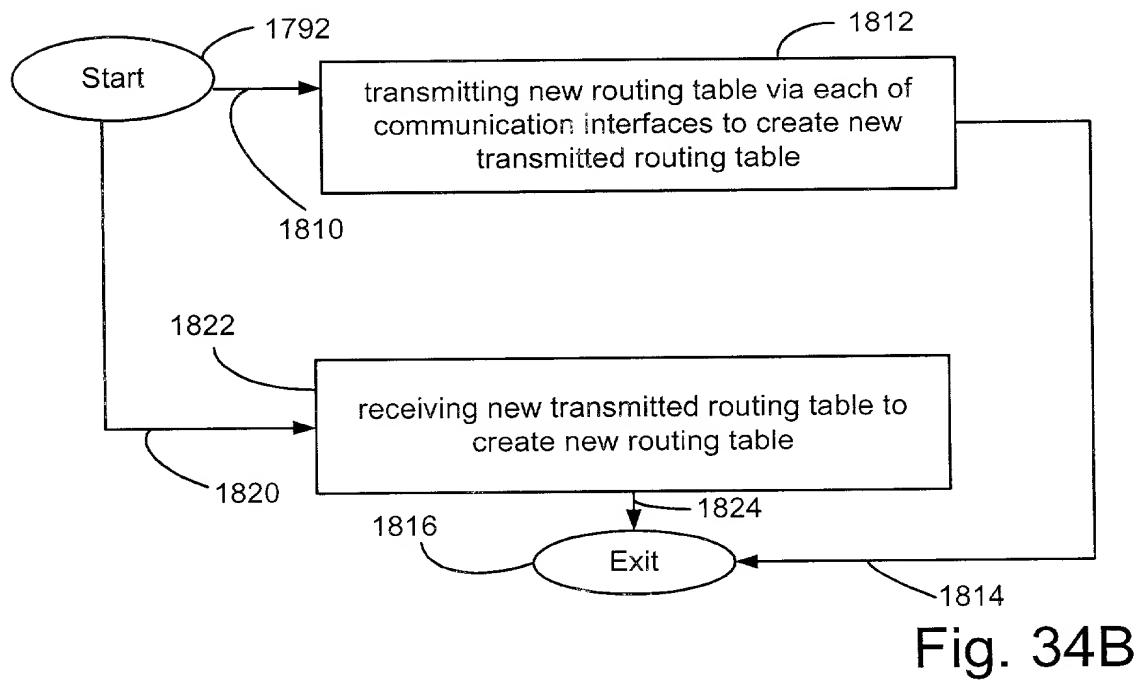
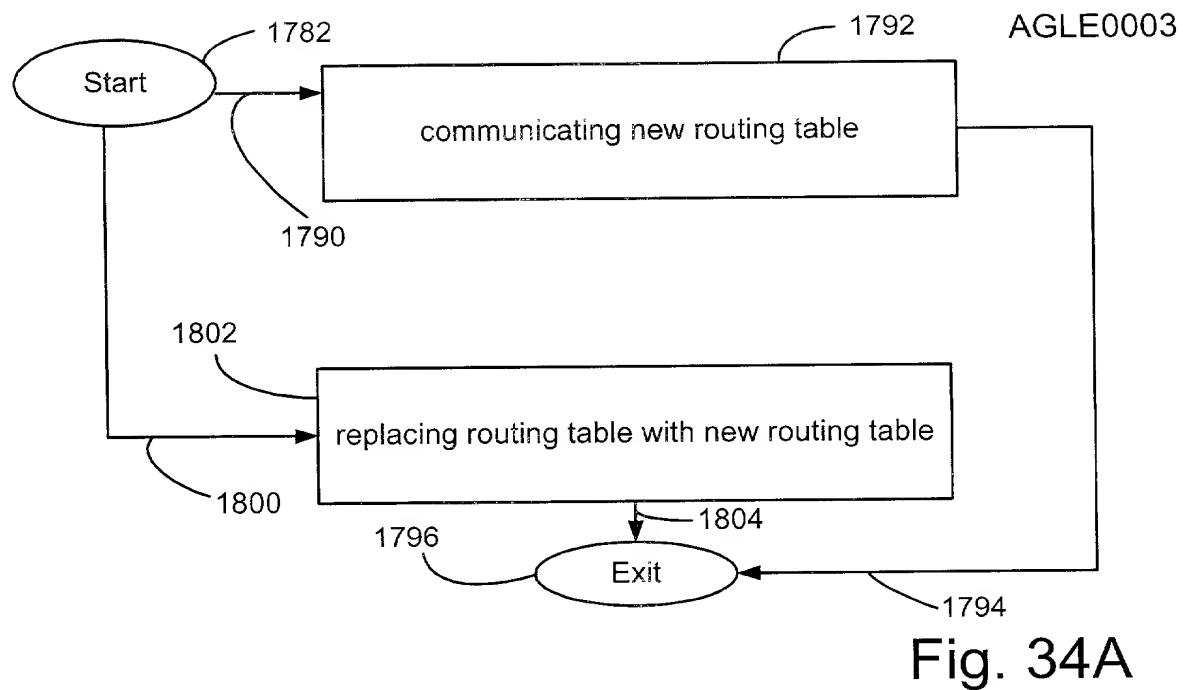


Fig. 33B



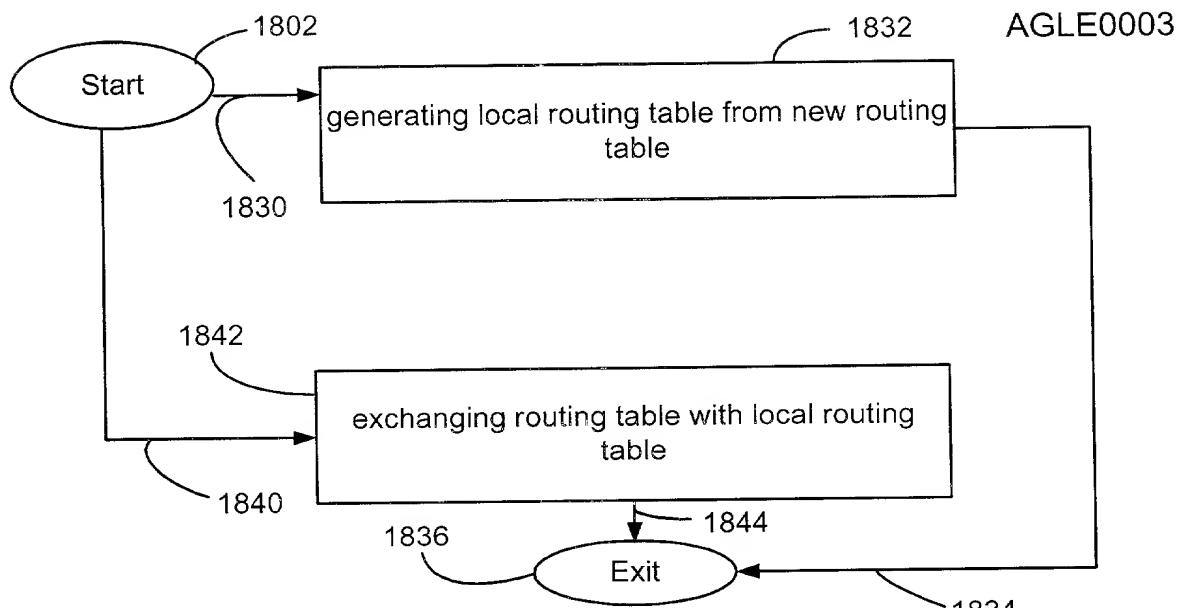


Fig. 35A

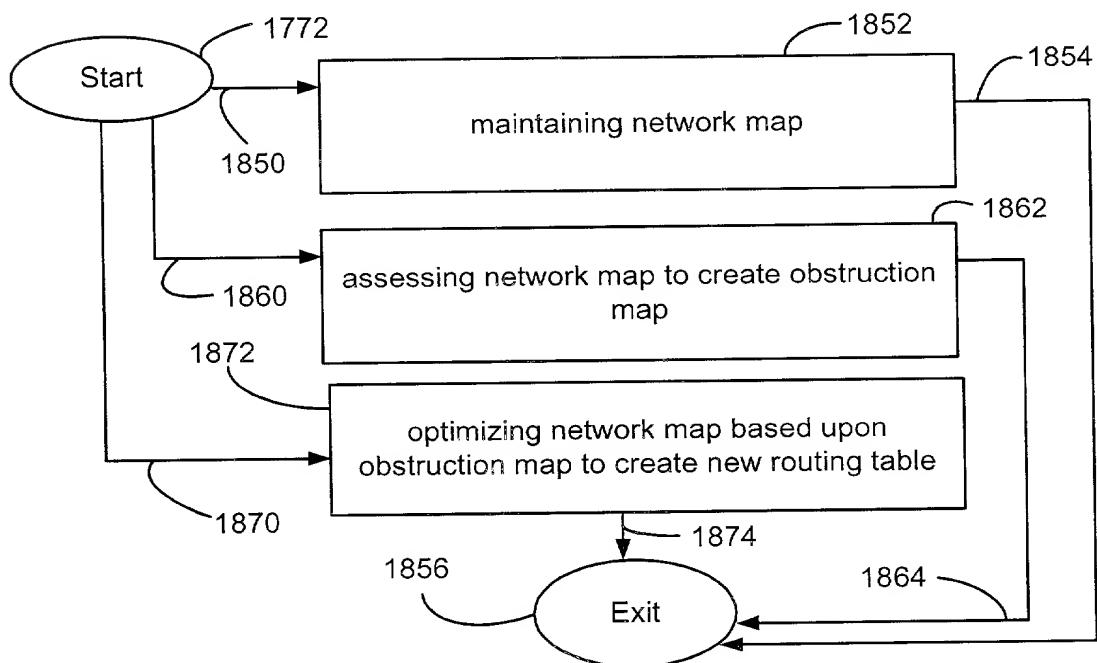


Fig. 35B

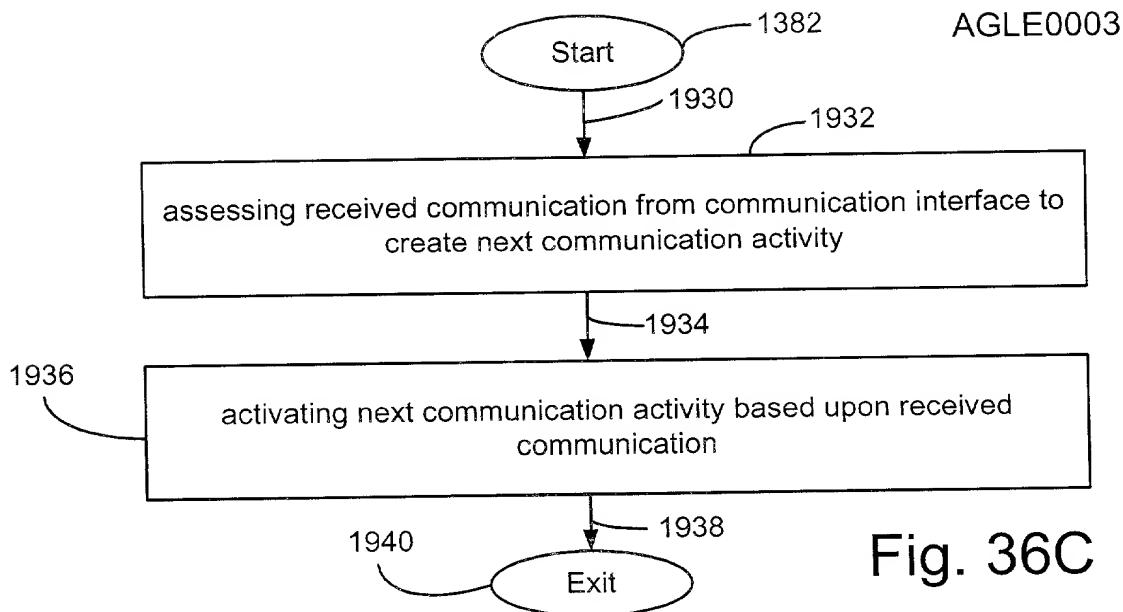


Fig. 36C

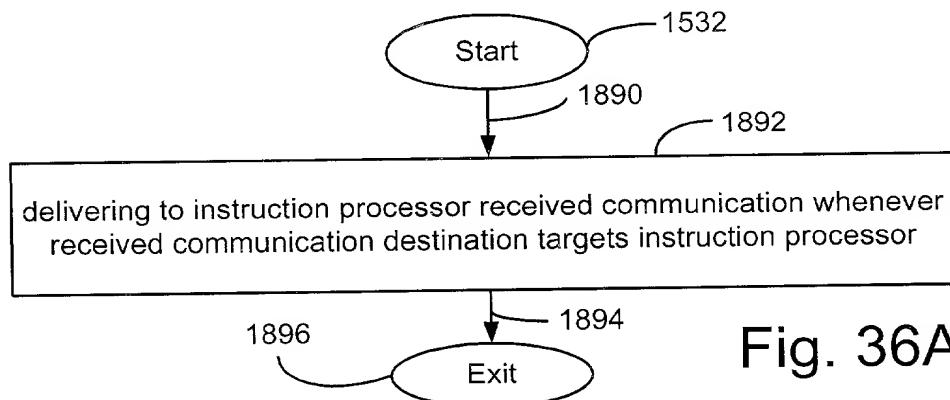


Fig. 36A

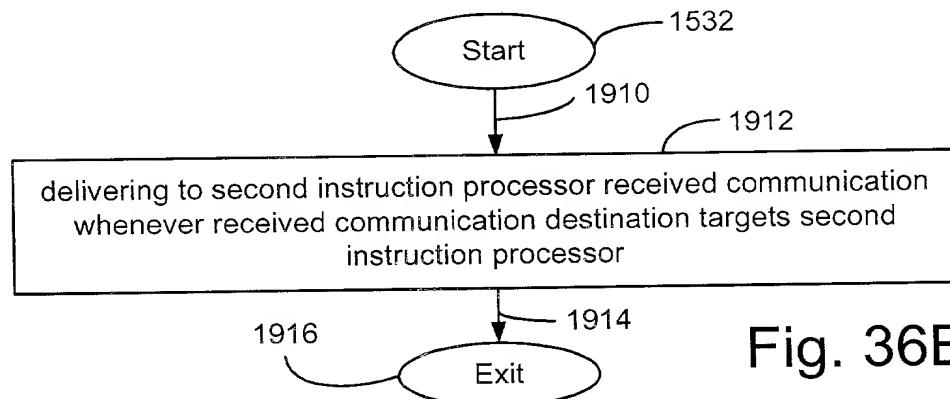


Fig. 36B

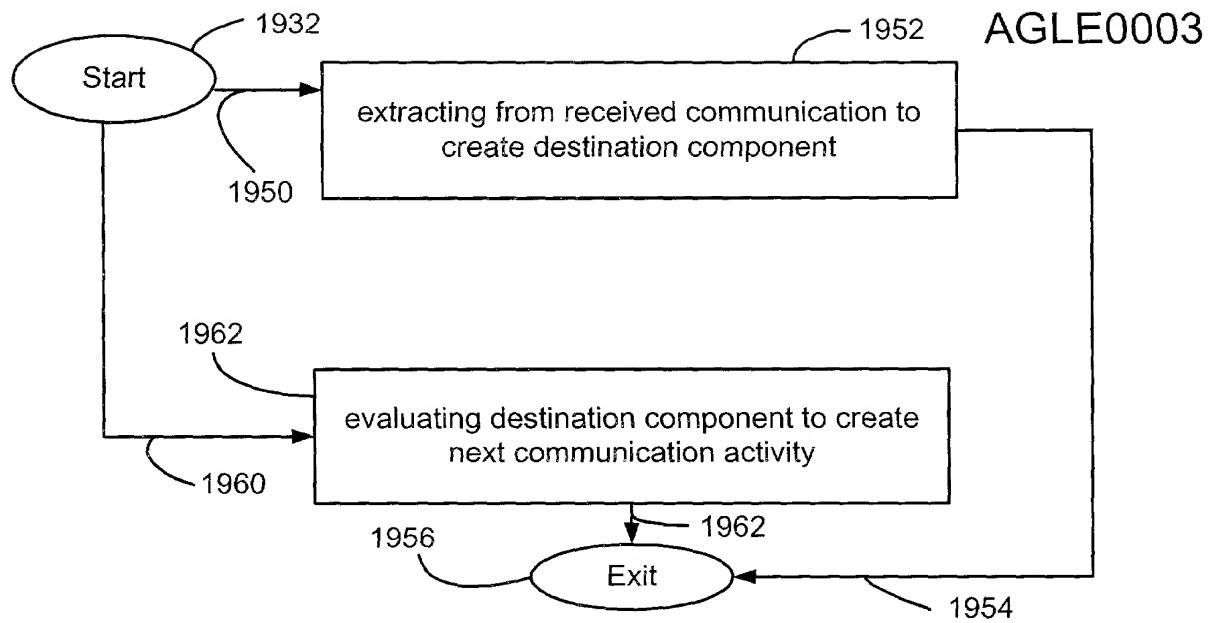


Fig. 37A

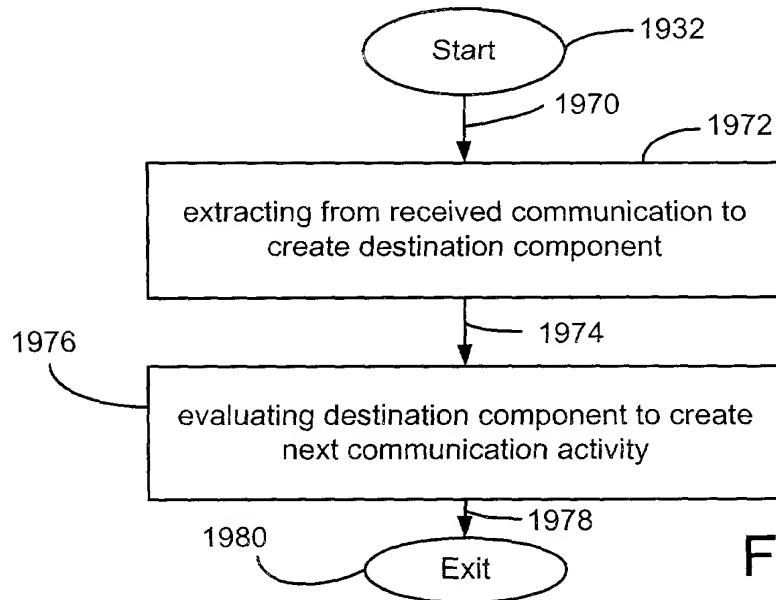


Fig. 37B

000626115-100400

AGLE0003

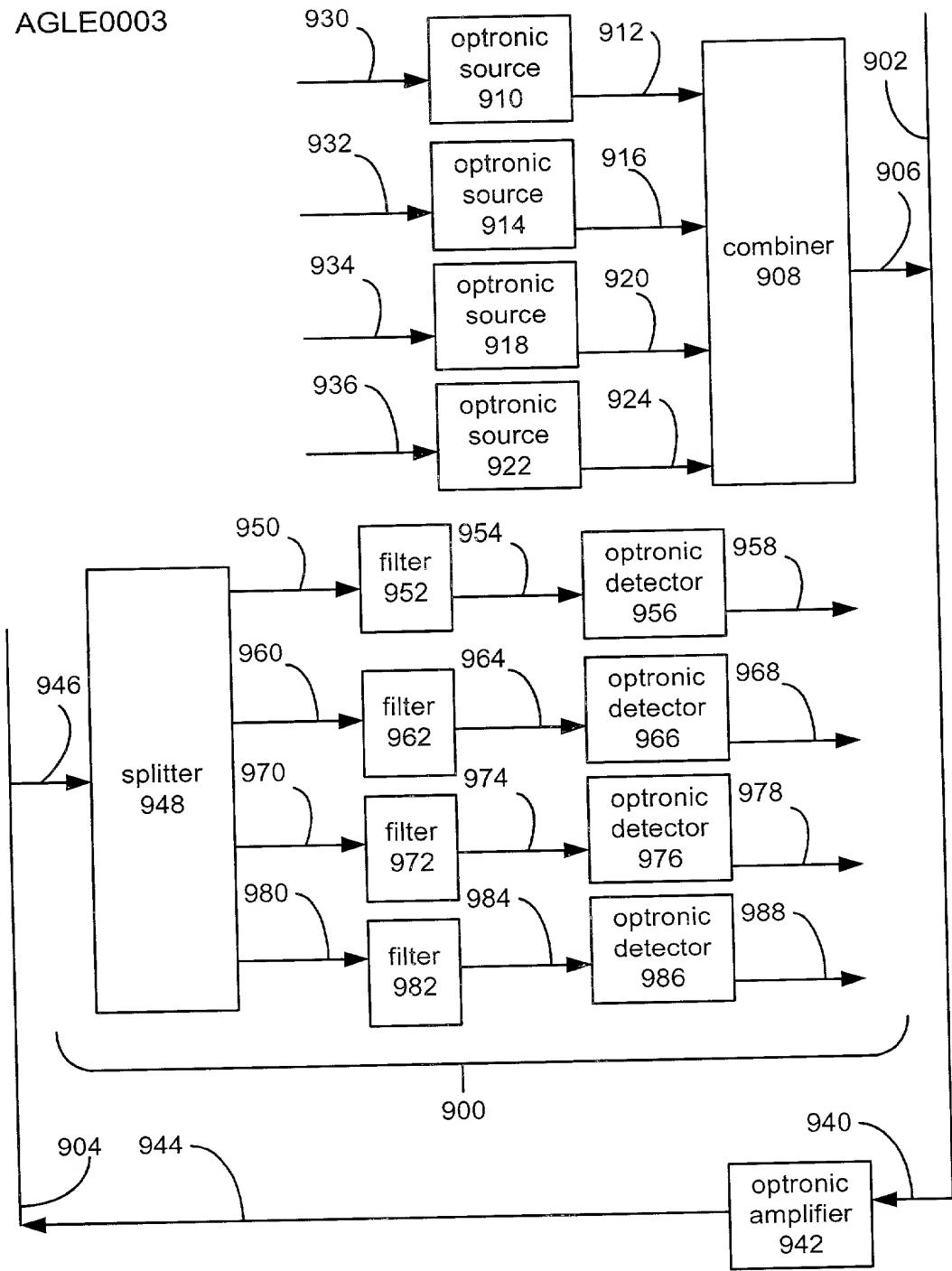


Fig. 38

DECLARATION FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name;

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

**SYSTEM AND METHOD OF A MULTI-DIMENSIONAL PLEX COMMUNIATION
NETWORK**

the specification of which (check one) X is attached hereto, or _____ was filed on _____ as Application Serial No. _____ and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

=====

I hereby claim foreign priority benefits under Title 35, United Sates Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)	Priority Claimed
	Yes No

Number Country Day/Month/Year Filed

Number Country Day/Month/Year Filed

=====

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

MICHAEL A. GLENN, Reg. No. 30,176

DONALD M. HENDRICKS, Reg. No. 40,355

CHRISTOPHER PEIL, Reg. No. 45,005

EARLE JENNINGS, Reg. No. 44,804

JACK J'MAEV, Reg. No. 45,669

SEND CORRESPONDANCE TO:

MICHAEL A. GLENN 3475 Edison Way, Suite L, Menlo Park, CA 94025

I hereby claim the benefit under Title 35, United States code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Ser. No. _____ Filing Date _____ Status: Patented, Pending, Abandoned _____

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor: THEODORE CALDERONE

Inventor's signature 

10/3/2000

Date

Residence 422 Portofino Drive, #3, San Carlos, California 94070

Post Office Address Same

Citizenship United States of America

Full name second inventor: MARK J. FOSTER

Inventor's signature 

10/3/00

Date

Residence 4157 Interdale Way, Palo Alto, California 94306

Post Office Address Same

Citizenship United States of America

```
*****  
/*----- PLEX ROUTER 0.4 -----*/  
/*  
 * AgileTV CONFIDENTIAL*  
 * Author: Mark J. Foster  
 * Last Updated: 09/27/00  
 */  
*****  
/* This module contains code to generate routing tables for the  
 * AgileEngine Plex architecture. This code is intended for three  
 * purposes:  
 */  
/*  
 * - Documenting the Plex architecture  
 * - Providing a simulation engine for analysis  
 * - Generating small, efficient routing tables  
 */  
/* Note that the code in this module would normally only be invoked  
 * at power-up, or when a component of the engine has failed, in  
 * which case it would be used to build a new routing table.  
 */  
/* In the actual system, the control CPUs will run similar code, ;  
 * and the resulting routing tables will be distributed to the  
 * other CPUs in the array using the PORT_BROADCAST method. This  
 * will cause each chip that receives the packet to retransmit it  
 * on all of its interfaces (other than the source port), thus  
 * ensuring that all nodes receive the routing table information,  
 * even in the presence of multiple link failures. To prevent  
 * these packets from being continuously retransmitted within the  
 * engine, the TTL (time to live) of each packet will be set to a  
 * small number - each retransmission decrements this counter, thus  
 * eventually ending the broadcast transmission.  
 */  
/* One of the most important functions of this code is to determine */  
/* the proper port (Ethernet port, LDT port, BRIDGE, TUNNEL, etc) */  
/* for each route, so that there is no run-time routing overhead.  
 */  
/* With this information, each chip in the array can perform */  
/* routing with a single int16 access to its unique routing table.  
 */  
/* This code also performs statistical failure analysis of the */  
/* Agile Engine, demonstrating its superior fault tolerance. As */  
/* one example, simulating 25 comm link failures in the core PLEX */  
/* arrays results in an average route length increase of just 8%!  
 */  
/* Inclusion of I/O demonstrates that the primary failure point */  
/* are the redundant I/O connection points, where the failure of */
```

jc825 U.S. PRO
09/679115



10/04/00

```

/*
 * an I/O processor, and its equivalent in the other PLEX plane
 */
/*
 * will result in a loss of connectivity.
 */
/*
 * Note that this code is written assuming the two chip/node
 * configuration, and that the PLEX arrays are powers of two...
 */
/*
 * The addressing scheme used here matches the "AgileTV Engine
 * Architecture" document, where rows are numbered from top to
 * bottom - 0..3 for PLEX 0 [left], 4..7 for PLEX 1 [right].
 */
/*
 * Columns are numbered from 0 [left] to 7 [right] within each PLEX.
 */
/*
 * Ethernet chips are numbered from 1..3, starting from the left.
 */
/*
 * Note that the CTRL chips actually do attach at diagonal points
 * within the array, as shown on the diagram, and are numbered
 * from 8.0 to 8.7, from left to right.
 */
/*
 * Note that the last six CHIPS in the array are actually not
 * real, but are dummy entries used to indicate external I/O lines.
 */
/*
 * By computing routing efficiency to these points, real-world
 * connectivity can be modelled in the presence of failures. In the
 * output produced by Print_Routes, the I/O lines are referred
 * to as indices 9.0..9.5, with 9.5 being the top-most of the
 * external I/O links shown on the upper left hand side of the
 * diagram.
***** */

/*
 * Change Log:
 */
/*
 * V0.4 - Removed "Ethernet through Bridge" and "Ethernet through
 * Bridge and LDT" transfer modes due to chip limitations.
 */
/*
 * Minor stats corrections, printing improvements.
 */
/*
 * V0.3 - Incorporated I/O line emulation
 */
/*
 * - Fixed several "invalid" failures
 */
/*
 * - Improved "bad route" detection
 */
/*
 * - Added more routing and coverage statistics
 */
/*
 * - Restructured stats for "run to failure"
 */
/*
 * V0.2 - Added statistics functions
 */
/*
 * - Code clean-up and documentation improved
 */
/*
 * V0.1 - Initial Release.
*/
***** */

/*
 * INCLURE FILE S
 */

```

```

/*
 *          P L E X   A R R A Y   C O N S T A N T S
 */
/* Load Windows pre-compiled headers */
/* Load standard library functions */
/* Load time functions */

/*
 *          S I M U L A T I O N   C O N S T A N T S
 */
/* Number of ROWS in the PLEX array */
/* Number of columns in the PLEX array (chips, not nodes) */
/* Number of CPUS per NODE - this is hard-coded throughout !!! */
/* Number of PLEX arrays in a single Agile Engine */
/* Each CHIP is a dual-processor SMP machine */
/* Number of Ethernet Links per chip */
/* Number of paths on a bridge chip (O<->C, E<->O) */
/* Number of "special" CTRL plus I/O chips per plane */
/* Total number of tunnels between PLEX planes */
/* Number of chips in a PLEX, offset to 2nd PLEX */
/* Mask for offset in PLEX array */
/* Number of external I/O connection points */
/* Number of LDT links between chips (w/o bridge) */
/* Total number of BRIDGE links in the machine */
/* Total number of CPU cores available for processing */
/* Total number of Ethernet links (/2 for src=dest) */

/*
 *          G E N E R A T I O N   C O R O S
 */
/* Number of times to run simulator */
/* Number of failures to inject into machine */
/* Maximum number of routing steps */
/* Number of route steps printed per line */
/* Insert newline for print format after N steps */
/* Insert second newline for LONG routes */

/*
 *          R O U T E S
 */
/* Generate a ROW from a comm address (chip number) */
/* Generate a MASKED ROW within this PLEX */
/* Generate a COLUMN from a comm address */
/* Get a node number for comparing adjacency */
/* Create a comm address from a (ROW, COL) pair */
/* Is the argument odd? ODD CPUS have row connections */
/* Is the argument even? Even CPUS have column connections */

```

```

/*
 *          D A T A   T Y P E S
 */
typedef unsigned char ADDR;
typedef unsigned char PORT;
typedef unsigned int COST;

/*
 *          C P U   S T R U C T U R E
 */
/* Define the communications address datatype */
/* Define the communications port datatype */
/* Datatype for routing cost */

/*
 *          C O R E
 */
/* Define a structure for an individual CPU core */
/* NOTE: This is not the same as CHIP, which has 2 cores */
/* CPU status: i.e., functional or not functional */
/* Eventually, we'll need to add load-sharing */

/* This CPU core is running and available */
/* Something went wrong! */
/* This CPU core isn't working */

/*
 *          R O U T E   S T R U C T U R E
 */
/* Define the structure for a route-table entry */
/* Target address (CHIP) this entry points to */
/* Port used to communicate to the target addr */
/* Cost of sending a packet along this complete route */
/* For efficiency, ROUTE will pack to 32 bits */

/*
 * Define values for Route.Port
 */
#define PORT_SELF 0
#define PORT_ETH_01 1
#define PORT_ETH_02 2
#define PORT_ETH_03 3
#define PORT_LDT 4
#define PORT_BTH 11 5
#define PORT_ETH_12 6
#define PORT_ETH_13 7
#define PORT_BRI_1 8
#define PORT_BRI_2 9
#define PORT_BRI_3 10
#define PORT_TUNNEL 11
#define PORT_EXT_10 12

/* Self-reference indicates end of routing - this CHIP is target */
/* Define Ethernet port 0 on this chip */
/* Define Ethernet 1 on current chip */
/* Ethernet 2 on current chip */
/* Define delta between Ethernets on this and other chip */
/* Define the LDT port for this chip */
/* Define Ethernet port 0 for other chip in this node */
/* Define Ethernet 1 on other chip in node */
/* Ethernet 2 on other chip in this node */
/* Xfer to "other" chip through bridge, or CTRL->EVEN */
/* Define transmission from CTRL to ODD */
/* Transmission to CTRL or I/O chip at bridged node */
/* "Tunneling" port between the PLEX arrays */
/* External Input or Output connection */

```

```

#define PORT_BROADCAST 13
#define PORT_NONE 99

/* Define values for Route.Cost */
#define COST_ETH 10
#define COST_LDT 5
#define COST_ETL 1
#define COST_BRIDGE 15
#define COST_ETB 18
#define COST_TUNNEL 15
#define COST_EXT_IO 25
#define COST_BIG 9999

/* Define special ADDR values */
#define ADDR_NONE 0xFF
#define ADDR_SELF 0xFE
#define ADDR_SPECIAL (PLEX * PLEXES)
#define ADDR_CTRL_0 (ADDR_SPECIAL + 0)
#define ADDR_IO_01 (ADDR_SPECIAL + 1)
#define ADDR_IO_02 (ADDR_SPECIAL + 2)
#define ADDR_IO_03 (ADDR_SPECIAL + 3)
#define ADDR_CTRL_1 (ADDR_SPECIAL + 4)
#define ADDR_IO_11 (ADDR_SPECIAL + 5)
#define ADDR_IO_12 (ADDR_SPECIAL + 6)
#define ADDR_IO_13 (ADDR_SPECIAL + 7)
#define ADDR_EXT_IO (ADDR_SPECIAL + 8)

/* Special flag for invalid route */
/* Special flag for Ethernet */
/* Assign routing cost for LDT */
/* Assign addl cost for route to "other" chip's Ethernet */
/* Cost of moving packets through LDT bridge chip */
/* Define cost of Ethernet through bridge */
/* Cost of "tunnel" connecting special CPUs */
/* Cost for external I/O */
/* Define LARGE cost - indicates no/broken link */

/* Special flag for invalid route */
/* Special flag for a self-reference (end of route) */
/* Address of the special chip group: CTRL + I/O CHIPS */
/* Control chip: PLEX array 0 */
/* IO chip Number 1 in PLEX array 0 */
/* IO chip Number 2 in PLEX array 0 */
/* IO chip Number 3 in PLEX array 0 */
/* Control chip: PLEX array 1 */
/* IO chip Number 1 in PLEX array 1 */
/* IO chip Number 2 in PLEX array 1 */
/* IO chip Number 3 in PLEX array 1 */
/* External I/O "dummy" chips */

/* **** C_H_I_P_S_T_R_U_C_T_U_R_E ****/
typedef struct _CHIP {
    ADDR addr;
    CORE core[NODE_SIZE];
    ROUTE route[CHIPS];
} CHIP;

/* **** S_T_A_T_S_S_T_R_U_C_T_U_R_E ****/
typedef struct _STAT {
    int routes;
    int bad;
    int min;
    int max;
    int avg;
    float pc;
} STAT;

```

```

} STATS;

/*
 *   F A I L U R E   S T R U C T U R E   *
 *   ****
 *   Define the structure for a failed link */
typedef struct _FAILURE {
    ADDR src;
    PORT dest;
    char *type;
} FAILURE;

/*
 *   Define the column-wise connectivity in the Agile Engine. Accessed by
 *   a source:destination column pair, this table contains the PORT
 *   needed to access the destination from the source.
 */

static PORT col_port [COLS] [COLS] =
{{PORT_SELF, PORT_LDT, PORT_ETH_03, PORT_ETH_02, PORT_ETH_01, PORT_ETH_01},
 {PORT_LDT, PORT_SELF, PORT_ETH_13, PORT_ETH_12, PORT_ETH_11, PORT_ETH_11},
 {PORT_ETH_01, PORT_ETH_01, PORT_SELF, PORT_LDT, PORT_ETH_03, PORT_ETH_02, PORT_ETH_02},
 {PORT_ETH_11, PORT_ETH_11, PORT_LDT, PORT_SELF, PORT_ETH_13, PORT_ETH_12, PORT_ETH_12},
 {PORT_ETH_02, PORT_ETH_02, PORT_ETH_01, PORT_ETH_01, PORT_SELF, PORT_LDT, PORT_ETH_03, PORT_ETH_03},
 {PORT_ETH_12, PORT_ETH_12, PORT_ETH_11, PORT_ETH_11, PORT_SELF, PORT_LDT, PORT_ETH_13, PORT_ETH_13},
 {PORT_ETH_03, PORT_ETH_03, PORT_ETH_02, PORT_ETH_02, PORT_ETH_01, PORT_ETH_01, PORT_SELF, PORT_LDT, PORT_ETH_14, PORT_LDT},
 {PORT_ETH_13, PORT_ETH_13, PORT_ETH_12, PORT_ETH_12, PORT_ETH_11, PORT_ETH_11, PORT_SELF, PORT_LDT, PORT_SELF}};

/*
 *   Define the row-wise connectivity in the Agile Engine. Accessed by a
 *   source:destination row pair, it returns the PORT connected to the
 *   destination. Unlike col_port, row_port can not contain the information
 *   necessary to identify when a transmission must occur across the
 *   other CPU in a node pair - this must be explicitly coded by
 *   examining the source's column number (i.e. if we are TXing from
 *   an EVEN CPU, it will be necessary to add PORT_LDT_OF to the values
 *   below).
 */

static PORT row_port [ROWS] [ROWS] =
{{PORT_SELF, PORT_ETH_01, PORT_ETH_02, PORT_ETH_03},
 {PORT_ETH_03, PORT_SELF, PORT_ETH_01, PORT_ETH_02},
 {PORT_ETH_02, PORT_ETH_03, PORT_SELF, PORT_ETH_01, PORT_ETH_01},
 {PORT_ETH_01, PORT_ETH_02, PORT_ETH_03, PORT_SELF, PORT_ETH_01},
 {PORT_ETH_01, PORT_ETH_02, PORT_ETH_03, PORT_SELF}};


```

```

/*
 * Define the physical index positions of the "special" I/O and CTRL CHIPS
 */
static ADDR phys_special[SPECIALS * PLEXES] = {ROWCOL(0,0), ROWCOL(1,2), ROWCOL(2,4), ROWCOL(3,6), ROWCOL(0,0) + PLEX, ROWCOL(1,2) + PLEX, ROWCOL(2,4) + PLEX, ROWCOL(3,6) + PLEX};

/*
***** G L O B A L E S ***** */
/* CHIP chip [CHIPS];          /* Array of physical CHIP descriptors */
/* CORE core [CORES];          /* Array of CPU core descriptors */
bool print;                   /* Global flag enabling printing of debug info */
int failures;                /* Number of failures we are simulating */
FAILURE failure [FAILS + 1];  /* Saved link failures for printing */

/*
***** random2: (int max, int *nl, int *n2) *****/
/* Random2 is called to generate a pair of pseudo-random integers
 * which are not equal to one another.
 */
/* Command line arguments:
 * max:      Numbers will range from 0..max-1
 * n1:      Pointer to the first integer variable
 * n2:      Pointer to the second integer variable
 */
/* Output:
 * NONE (*nl and *n2 set to 0..max-1)
 */
void random2(int max, int *nl, int *n2)
{
    *nl = rand() % max;
    do {
        *n2 = rand() % max;
        } while (*nl == *n2);
    } /* random2 */

/*
***** print_route: (ROUTE *route) *****/
/* Print_Route is called to print a single route step in a human
 * readable fashion.
 */
/* Input:
 */

```

```

/*
 * route: Pointing to the desired ROUTE structure
 */
/*
 * Output:
 */
/* STDOUT
 * Returns TRUE if this path has more route steps
 */
***** Print ROUTE *route
{
    char *port_name[] = {"SELF", "ETH [1]", "ETH [2]", "ETH [3]",
                         "LDT [-]", "ETL [1]", "ETL [2]", "ETL [3]",
                         "BRI [1]", "BRI [2]", "BRI [3]", "TUNNEL",
                         "I/O [-]", "BROADCAST"};
    switch (route->port) {
        case PORT_SELF: /* Format output based on routing type */
            /* Source = dest: end of route */
            /* Reached end of communications path */
            /* No route available to the destination */
            /* End of this communication path */
            /* Data transfer (ETH, LDT, BRIDGE, etc.) */
            break;
        case PORT_NONE:
            printf("**** No Route Available ****");
            return (false);
            break;
        default:
            printf("%s %x.%x ", port_name[route->port], ROW(route->addr));
            break;
    }
    return (true);
} /* print_route */

/*
 * print_routes: (start, n, stats, path)
 */
/* Print_Routes is used for debugging. It will print out the
 * specified rectangular region of the routing tables, optionally
 * tracing the complete path through the engine.
 */
/* Input:
 *   start: Starting chip number
 *   chips: Number of chips to print
 *   stats: Pointer to STATS for this routing table
 *   path: Set to TRUE to print complete routing path
 */
/* Output:
 *   STDOUT
 */
***** Print ROUTE *route(ADDR start, int n, STATS stats, bool path)
void print_routes(ADDR start, int n, STATS stats, bool path)
{
    ADDR src, dest, here; ROUTE *route; int i, time_to_live;
    for (i = 0; i < failures; ++i) {
        printf("DELETE %s: %x.%x->%x.%x\n", failure[i].type, /* First, print out simulated link failures */
               failure[i].src, /* Print out the stored table */
               ROW(failure[i].src), /* ...of src->dest failures */
               failure[i].src);
    }
}

```

```

ROW(failure[i].dest) , COL(failure[i].dest)) ;

    for (src = start; src < (start + n); ++src) { /* For each source chip */
        route = &chip[src].route[dest]; /* For each possible destination */
        if (route->port != PORT_NONE) {
            printf("%x.%x.%x.%x (%3d) - ", ROW(src), COL(src), ROW(dest), COL(dest), route->cost);
            here = src;
            if (here == dest) printf("DONE\n");
            else {
                while (here != dest) {
                    route = &chip[here].route[dest]; /* Point to the route */
                    if (!print_route(route)) break; /* Have found a valid route */
                    if (here->time_to_live) break; /* Set maximum number of routing hops */
                    here = route->addr;
                    if (here != dest) && (time_to_live==LINE_BREAK || time_to_live==LINE_BREAK2) {
                        printf("\n\t\t");
                        if (here==dest) printf("..."); /* Discovered bad route: too many steps */
                        else printf("..."); /* Advance to next routing step */
                    }
                    if (here == dest) && (time_to_live==LINE_BREAK || time_to_live==LINE_BREAK2) {
                        printf("\n\t\t");
                        if (here==dest) printf("..."); /* If we have more route steps to print... */
                        else printf("..."); /* ...and reached end of first line... */
                    }
                }
                if ((here==dest) || time_to_live) { /* Reached end of this route? */
                    printf("\n");
                    else printf("[TERMINATED]\n");
                }
            }
        }
    }
}

printf("Coverage: %2.1f%%, " , stats.pc); /* Print percent of engine reachable */
printf("Routes: %4d, " , stats.routes); /* Print total number of routes */
printf("Bad: %d, " , stats.bad); /* Print number of special routes */
printf("Min: %d, " , stats.min); /* Print minimum path length */
printf("Max: %d, " , stats.max); /* Print maximum path length */
printf("Avg: %d\n" , stats.avg); /* Print average route length */
} /* print_routes */

***** add_route: (ADDR src, ADDR dest, PORT port, COST cost)
/*
 * Add_route is called to add a new route to the table. Note that
 * unlike delete_route, add_route does not set up the reverse path
 * automatically, since different port numbers and costs are
 * associated with each direction of the route.
 */

```

```

/* Inputs:
/*   src:      Source communications address (CHIP number) */
/*   dest:     Destination communications address (CHIP number) */
/*   port:    Port number to use to transmit along this route */
/*   cost:    Cost of taking this routing step */
/* Output:
/*   NONE
/****************************************************************************/
void add_route(ADDR src, ADDR dest, PORT port, COST cost)
{
    chip[src].route[dest].addr = dest;                                /* Set the destination address */
    chip[src].route[dest].port = port;                                    /* Store the port to send this to */
    chip[src].route[dest].cost = cost;                                    /* Save the cost of this route */
} /* add_route */

/****************************************************************************/
/* delete_route: (ADDR src, ADDR dest)
*/
/* Delete_Route is called to delete a single communications route
/* from the PLEX array. Note that this routine will delete the
/* forward and reverse paths for the specified route.
*/
/* Input:
/*   src:      Source communications address (CHIP number)
/*   dest:     Destination communications address (CHIP number)
/* Output:
/*   NONE
/****************************************************************************/
void delete_route(ADDR src, ADDR dest)
{
    chip[src].route[dest].cost = COST_BIG;                                /* Make this route too expensive to traverse */
    chip[src].route[dest].port = PORT_NONE;                                /* Flag this as end of route */
    chip[dest].route[src].cost = COST_BIG;                                /* Delete the reverse route, as well */
    chip[dest].route[src].port = PORT_NONE;                                /* Set reverse link as end route */
} /* delete_route */

/****************************************************************************/
/* fail_link: (ADDR src, ADDR dest, char *type)
*/
/* Fail_Link is called to delete a complete communications link
/* from the PLEX array. This routine is called when one of the
/* physical Ethernet or busses has failed. If the link is an
/* Ethernet link, the routine will also delete the routes from the
/* other CPUs in the source and destination nodes. If the link is
/* an LDT bus, the routine will delete all of the communication
/* routes which are no longer usable as a result: all of the row
*/

```

```

/* routes to the EVEN CPU, and all of the column routes to the ODD */
/* CPU. If failing a link to a special CPU (CTRL or I/O), 'src' */
/* should contain the address of the special chip, and dest the */
/* address in the PLEX array (unless dest is also a special chip, */
/* in which case this is a TUNNEL failure). If failing a link to */
/* an I/O, src must contain the index of the I/O. */
*/
/*
 * Input:
 *   src:      Source communications address (CHIP number)
 *   dest:     Destination communications address (CHIP number)
 *   type:    Pointer to string containing failure name
 */
/*
 * Output:
 *   NONE
 */
void fail_link(ADDR src, ADDR dest, char *type)
{
    int i;
    delete_route(src, dest);
    if (src < ADDR_SPECIAL) {
        if ((src ^ 1) != dest) {
            delete_route(src ^ 1, dest);
            delete_route(src, dest ^ 1);
            delete_route(src ^ 1, dest ^ 1);
        }
    }
    else {
        for (i = 0; i < COLS; ++i) {
            if (i != COL(dest)) {
                delete_route(dest, ROWCOL(ROW(dest), i));
            }
        }
    }
    for (i = 0; i < ROWS; ++i) {
        if (i != ROW(src)) {
            delete_route(src, ROWCOL(i, COL(src)));
        }
    }
}
} /* fail_link */

/*
 * fail_random_links (int fails)
 */
/*
 * Fail_Random_Links is called to generate a specified number of
 * communications link failures within the Agile Engine. Under the
 * assumption that all links are equally likely to fail, this
 * routine will generate statistically correct failures within
 * each component type (Ethernet, BRIDGE, LDT, TUNNEL).
 */

```

```

/*
 * Inputs:
 *   fails:  Number of links to fail
 */
/*
 * Outputs:
 *   NONE
 */
******/
```

```

void fail_random_links(int fails)
{
    int src, dest, tmp; char *type;
    for (failures=0; failures < FAILS; ++failures) { /* Simulate this many failures */
        tmp=rand()%(ETHS+BRIDGES+LDTs+TUNNELS+EXT_IOS); /* Generate a link number */
        /* Fail an external I/O connection
        */
        if (tmp >= (ETHS + BRIDGES + LDTs + TUNNELS)) { /* Time to fail an external I/O connection */
            src = rand() % EXT_IOS; /* Figure out which I/O is failing */
            dest = ADDR_SPECIAL + (src >> 1) + 1; /* Point to the proper CTRL chip */
            if (rand() & 1) dest += SPECIALS; /* Toss a coin: PLEX 0 or PLEX 1 */
            src += ADDR_EXT_IO; /* Point to the I/O line's "CHIP" */
            type = "I/O"; /* Set the failure type */
        }
        /* Fail a TUNNEL link between PLEX planes
        */
        else if (tmp >= (ETHS + BRIDGES + LDTs)) { /* Time to fail a TUNNEL */
            src = (rand() % SPECIALS)+ADDR_SPECIAL; /* Get the number of a special chip */
            dest = src + SPECIALS; /* Point to connection in other PLEX */
            type = "TUNNEL"; /* Set the failure type */
        }
        /* Fail an LDT link
        */
        else if (tmp >= (ETHS + BRIDGES)) { /* An LDT bus link has failed */
            do { /* Make sure that this isn't a BRIDGE location */
                src = (rand() % PLEX) & ~1; /* Find an even chip within the PLEX */
                } while (ROW(src) == COL(src * 2)); /* Oops - ran into a BRIDGE! */
                if (rand() & 1) src += PLEX; /* Place in second PLEX 50% of the time */
                dest = src ^ 1; /* Point to the other half of the plane */
                type = "LDT"; /* Get name of failed link */
            }
            /* Fail an LDT Bridge chip
            */
            else if (tmp >= ETHS) { /* We're about to fail a BRIDGE link */
                src = rand() % (SPECIALS * PLEXES); /* Find one of the special chips */
                tmp = rand() % BRIDGES_CHIP; /* Find which bridge patch has failed */
                if (tmp) dest = phys_special[src] ^ 1; /* This failure is to the ODD chip */
                else dest = phys_special[src]; /* This failure is to the EVEN chip */
            }
        }
    }
}

```

```

if (tmp< (BRIDGES_CHIP-1) ) src+=ADDR_SPECIAL; /* This failure is from a CTRL chip */
else src = phys_special[src]; /* This failure is from an EVEN chip */
type = "BRIDGE";
}

/*
 * Fail a column connection in the PLEX array
 */
else if (tmp >= (ETHS / 2)) {
    do {
        tmp = rand() % CHIPS;
        tmp &= ~(COLS - 1);
        random2(COLS, &src, &dest);
        src = (src + tmp) & ~1;
        dest = (dest + tmp) & ~1;
        while (src == dest);
        type = "COLUMN";
    }
}

/*
 * Fail a row connection in the PLEX array
 */
else {
    random2(ROWS, &src, &dest);
    tmp = (rand() % COLS) | 1;
    if (rand() & 1) {
        src += ROWS;
        dest += ROWS;
    }
    src = ROWCOL(src, tmp);
    dest = ROWCOL(dest, tmp);
    type = "ROW";
}

fail_link(src, dest, type);
failure[failures].src = src;
failure[failures].dest = dest;
failure[failures].type = type;
}

} /* fail_random_links */

*****route_stats: (smin, slen, dmin, dlen, *stats)
/*
 * Statistics is called to generate statistics for the current
 * routing table. It will generate information about the total
 * number of routes, and the average length of each route.
 * The routine will produce statistics for the source range starting
 * with "smin" of length "slen", connecting to destinations from
 * "dmin" for a range of "dmax".
 */

```

```

/*
 *  Inputs:
 *    smin: Starting source chip index
 *    slen: Number of source chips to check
 *    dmin: Starting destination chip index
 *    dest: Number of dest chips to check
 *    stats: Pointer to STATS structure for results
 */
/* Output:
 *    NONE
 */
void route_stats(ADDR smin, ADDR slen, ADDR dmin, ADDR dlen, STATS *stats)
{
    ADDR src, dest; ROUTE *route; int here, cost, time_to_live;
    stats->routes = 0;
    stats->min = COST_BIG;
    stats->max = 0,
    stats->bad = 0;
    int route_length = 0;
    for (src = smin; src < (smin + slen); ++src) {
        for (dest = dmin; dest < (dmin + dlen); ++dest) {
            route = &chip[src].route[dest];
            cost = route->cost;
            ++stats->routes;
            here = src;
            if (route->port != PORT_NONE && cost) {
                route_length += cost;
                if (cost < stats->min) stats->min=cost;
                if (cost > stats->max) stats->max=cost;
                time_to_live = MAX_ROUTES;
                while (here != dest) {
                    route = &chip[here].route[dest];
                    if (route->port == PORT_NONE) break;
                    here = route->addr;
                    if (!--time_to_live) break;
                }
                if (here != dest) ++stats->bad;
            }
        }
    }
    stats->avg = route_length / stats->routes;
    stats->pc=100.0-(100.0*stats->bad)/stats->routes;
} /* route_stats */
}

/*
 *  compute_routes: (
 */
/* Compute_percent_route_coverage */
/* Compute_Routes is called once the routing structures have */

```

```

/*
 * been initialized, and all engine connections have been added, */
 * and all known failures have been recorded. Once this routine is */
 * completed, the routing arrays will contain the shortest path */
 * from every source to every destination, and the cost of each */
 * route step will contain the <total> route cost for the path it */
 * points to.
*/
/*
 * This routine implements Least Cost Routing, which is another way */
 * of referring to the "All Pairs Shortest Path" problem. The */
 * algorithm used below is known as Floyd's algorithm (for details, */
 * see the book Data Structures and Algorithms, by Aho, Hopcroft, */
 * and Ulman).
*/
/*
 * Inputs:
 * CHIP and ROUTE arrays initialized with physical ROUTES and */
 * route costs.
*/
/*
 * Output:
 * Fully optimized routing tables!
*/
******/
```

```

void compute_routes()
{
    ADDR src, dest, target; COST newcost; ROUTE *route;
    for (target=0; target < ADDR_EXT_IO; ++target) {
        /* For all valid intermediate routes */
        for (src=0; src < CHIPS; ++src) {
            /* For every possible CCMW source */
            for (dest = 0; dest < CHIPS; ++dest) {
                /* ...to every possible destination */
                /*
                 * Prevent "loopbacks" through external interfaces. It is
                 * not acceptable to send packets to the outside world as
                 * a method of communicating between PLEX planes!
                */
                if ((src >= ADDR_EXT_IO) && (dest >= ADDR_EXT_IO)) continue;
                /* See if the addition of a new routing step will be "less
                 * expensive" than the current route for this step. This
                 * is done by comparing the costs of the current route to the
                 * sum of: the cost from the source to the intermediate node
                 * plus the cost of the intermediate node to the destination.
                 * If this new cost is less, point this node to the new
                 * intermediate node, and set the path cost to the summed
                 * cost through the intermediate node.
                */
                route = &chip[src].route[dest];
                newcost=chip[src].route[target].cost +
                        chip[target].route[dest].cost;
                if (newcost < route->cost) {
                    /* If new cost is less, reroute through intermediate */
                    route->addr = chip[src].route[target].addr; /* Get path to the new destination */
                    route->port = chip[src].route[target].port; /* Copy the port from the path to this location */
                    route->cost = newcost;
                }
            }
        }
    }
}
```

```

    }

} /* compute_routes */

/*
 * init_special: (ADDR chip)
 */
/*
 * Init_Special is called to initialize a "special" I/O or CTRL
 * chip. This routine will attach the control chip to its
 * physical location in the PLEX array, and will generate routes
 * for the chip. Note that this routine will also handle the
 * increased cost associated with the LDT bridge chip located at
 * the PLEX locations where the special chips are located. This
 * routine must only be called <after> the basic PLEX arrays have
 * been initialized, but prior to deletion of any failed routes!
 */
/*
 * Inputs:
 *   chip: Contains the index of the special chip to be added
 */
/*
 * Output:
 *   NONE
 */
void init_special(ADDR chip)
{
    ADDR phys;
    phys = phys_special[chip - ADDR_SPECIAL];
    /* Get this chip's location in PLEX array */
    /* Initialize this chip's end of route */
    /* Add route from CTRL to EVEN */
    /* ... and reverse path from EVEN to CTRL */
    /* Add route from CTRL to ODD */
    /* ... and reverse path from ODD to CTRL */
    /* ... and reverse path from ODD to CTRL */
    /* Set higher routing cost for existing node */
    /* ... and for the reverse bridge path */
} /* init_special */

/*
 * init_route: (ADDR src, ADDR dest)
 */
/*
 * Init_Route is called to initialize a PLEX array entry for the
 * main CPU arrays. This routine will initialize the route cost,
 * address, and port for this physical link. Note that this
 * routine must not be used for the CTRL or I/O CPUs, since they
 * require special handling!
 */
/*
 * Input:
 *   src:      Source communications address (CHIP number)
 */

```

```

/*
 *      dest: Destination communications address (CHIP number)
 */
/* Output:
 *      NONE
***** */
void init_route(ADDR src, ADDR dest)
{
    PORT port; COST cost;
    if (ROW(src) == ROW(dest)) {
        port = col_port[COL(src)][COL(dest)];
        if (port == PORT_SELF) cost = 0;
        else if (port == PORT_LDT) cost = COST_LDT;
        else {
            cost = COST_ETH;
            if (ODD(src)) cost += COST_ETL;
            if (ODD(dest)) cost += COST_ETL;
        }
        add_route(src, dest, port, cost);
    }
    else if (NODE(src) == NODE(dest)) {
        port = row_port[ROW_MASK(src)][ROW_MASK(dest)];
        cost = COST_ETH;
        if (EVEN(src)) {
            port += PORT_LDT_OF;
            cost += COST_ETL;
        }
        if (EVEN(dest)) cost += COST_ETL;
        add_route(src, dest, port, cost);
    }
} /* init_route */

/*
 * init_routes: ()
 */
/* Init_Routes is called to initialize the routing table prior to
 * link failures or optimizations. It adds all of the physical
 * elements of the table, starting with the PLEX arrays, then it
 * adds the special CTRL and I/O CPUs, then finally the tunnels
 * between the PLEX arrays at the special chip locations. After
 * init_routing is completed, any array failures should then be
 * incorporated, then finally, the routing table may be computed.
 */
/* Inputs:
 *      NONE
 */
/* Output:
 *      NONE
***** */

```

```

void init_routes()
{
    ADDR_src, dest; int i;
    for (i = 0; i < CORES; ++i) {
        core[i].status = OK;
        core[i].workload = 0;
    }
    for (src = 0; src < CHIPS; ++src) {
        for (dest = 0; dest < CHIPS; ++dest) {
            chip[src].route[dest].port = PORT_NONE;
            chip[src].route[dest].cost = COST_BIG;
        }
    }
    for (src = 0; src < PLEX; ++src) {
        for (dest = 0; dest < PLEX; ++dest) {
            init_route(src, dest);
            init_route(src + PLEX, dest + PLEX);
        }
    }
    for (i = 0; i < SPECIALS * PLEXES; ++i) {
        init_special(ADDR_SPECIAL + i);
    }
    for (i = 0; i < SPECIALS; ++i) {
        src = ADDR_SPECIAL + i;
        add_route(src, src, PORT_SELF, 0);
        dest = ADDR_SPECIAL + SPECIALS + i;
        add_route(src, dest, PORT_TUNNEL, COST_TUNNEL);
        add_route(dest, src, PORT_TUNNEL, COST_TUNNEL);
    }
    for (i = 0; i < EXT_IOS; ++i) {
        src = ADDR_EXT_IO + i;
        add_route(src, src, PORT_NONE, 0);
        dest = ADDR_SPECIAL + (i >> 1) + 1;
        add_route(src, dest, PORT_EXT_IO, COST_EXT_IO);
        add_route(dest, src, PORT_EXT_IO, COST_EXT_IO);
        dest += SPECIALS;
        add_route(src, dest, PORT_EXT_IO, COST_EXT_IO);
        add_route(dest, src, PORT_EXT_IO, COST_EXT_IO);
    }
} /* init_routes */

/*
 * main: ()
 */
/* This program generates routing tables for the Agile Engine,
 * and simulates engine performance in the presence of failures.
 * It first generates optimal routes for a functional engine,
 * printing out the average route length. Then, it injects random */

```

```

/*
 * failures of the communications paths in the machine, generating */
 * new routing tables, and will print the final average route */
 * length (usually quite close to the no-error case!). The program */
 * will also print out the complete route tables for the last */
 * simulation run.
 */
/*
 * Command line arguments:
 */
/*      NONE */
/*
 * Output:
 */
/*      STDOUT */
***** */

int main(int argc, char* argv[])
{
    long pass; STATS stats;
    srand((unsigned)time(NULL));
    init_routes();
    compute_routes();
    for (pass = 0; pass < PASSES; ++pass) {
        print = (pass == (PASSES - 1));
        init_routes();
        fail_random_links(FAILS);
        compute_routes();
        route_stats(ADDR_EXT_IO, EXT_IOS, 0, CHIPS-EXT_IOS, &stats);
        if (stats.bad) {
            /* Found part of engine we can't talk to */
            printf("!!! FAILURE! Pass: %ld, Simulated Failures: %d, BAD ROUTES: %d ***\n", pass, FAILS, stats.bad);
            print_routes(0, CHIPS, stats, true);
            print = true;
            break;
        }
    }
    if (!stats.bad) print_routes(0, CHIPS, stats, true); /* Print the global routing table */
    return (OK);
} /* main */

```